

## THESIS / THÈSE

### MASTER EN SCIENCES INFORMATIQUES

Dynamique des mécanismes d'échange entre nœuds d'un réseau à commutation de paquets

Sainvitu, Jean-Pierre

*Award date:*  
1978

*Awarding institution:*  
Université de Namur

[Link to publication](#)

#### General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal ?

#### Take down policy

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

**FACULTES UNIVERSITAIRES NOTRE - DAME DE LA PAIX  
INSTITUT D'INFORMATIQUE  
NAMUR**

**DYNAMIQUE DES MECANISMES  
D'ECHANGE ENTRE NOEUDS  
D'UN RESEAU A COMMUTATION DE PAQUETS**

**Mémoire présenté par  
Jean-Pierre SAINVITU  
en vue de l'obtention du grade  
de Licencié et Maître en  
Informatique**

**Année Académique  
1977 - 1978**



FNB 16 | 1978 | 1

FACULTES  
UNIVERSITAIRES  
N.-D. DE LA PAIX  
NAMUR

Bibliothèque

FNB

16/1378/1

**FACULTES UNIVERSITAIRES NOTRE - DAME DE LA PAIX  
INSTITUT D'INFORMATIQUE  
NAMUR**



**DYNAMIQUE DES MECANISMES  
D'ECHANGE ENTRE NOEUDS  
D'UN RESEAU A COMMUTATION DE PAQUETS**

**Mémoire présenté par  
Jean-Pierre SAINVITU**  
en vue de l'obtention du grade  
de Licencié et Maître en  
Informatique

**Année Académique  
1977 - 1978**





99030

Ce mémoire a été réalisé au cours d'un stage au département Systèmes et Automatique de la Faculté des Sciences Appliquées de l'Université de Liège, dirigé par Mr. le Professeur A. Danthine.

Je remercie Mr. Brunin, Professeur à l'Institut d'Informatique de Namur, promoteur de ce mémoire, et Mr. Danthine qui ont déterminé l'orientation du travail.

Je remercie tout particulièrement Mr. Patrice Ksrewinski, Assistant au département Systèmes et Automatique de l'Université de Liège, dont les suggestions et les critiques m'ont été très utiles tout au long de l'élaboration du travail.

Je remercie également toute l'équipe des chercheurs de ce département pour leurs conseils.



PRELIMINAIRES1.1. Les réseaux d'ordinateurs à commutation de paquets.

Avec l'utilisation croissante des ordinateurs, un grand intérêt s'est porté sur la possibilité d'interconnection de centres d'ordinateur pour former des réseaux. De nombreux secteurs, tels que les banques, les systèmes de réservation des lignes aériennes, demandent des moyens de transmission de données efficaces pour pouvoir coordonner leurs activités dispersées géographiquement. L'idée du partage de ressources entre plusieurs centres sous la forme d'un réseau mène à l'élimination d'information redondante: chaque centre ne doit pas conserver ses propres copies de programmes ou de données particulières; de plus, un plus grand volume d'information, de données et de systèmes "hardware" spécialisés est disponible pour chaque utilisateur; chaque centre peut par exemple ne posséder qu'un nombre restreint de compilateurs.

Des systèmes incompatibles à plusieurs points de vue (les structures de données, les méthodes d'accès,...) peuvent être interconnectés dans un réseau s'ils acceptent un ensemble commun de conventions d'interface pour le hardware et pour les données.

Un réseau d'ordinateur est un ensemble de processeurs interconnectés qui peuvent être utilisés conjointement mais qui sont contrôlés par des systèmes d'exploitation séparés et qui peuvent travailler d'une façon autonome.

Voyons maintenant quelles sont les différentes méthodes de commutation. Si, dans un réseau téléphonique, chaque utilisateur était relié par une ligne à chaque autre utilisateur, la plupart des lignes seraient inutilisées la majorité du temps. On utilise plutôt des centres de commutation, auxquels tous les utilisateurs sont reliés et qui permettent de connecter l'appelé et l'appelant. On établit un chemin de communication constitué de lignes libres entre la source et la destination pour toute la durée de l'appel, que la conversation ait lieu ou pas: cette technique s'appelle la commutation de circuit. Si cette méthode s'avère efficace et



rentable pour la transmission continue de volumes importants de données, elle ne l'est pas dans les autres cas. Les alternatives de la commutation de circuit sont la commutation de message ou la commutation de paquet. Dans la commutation de message, il n'y a pas de connexion physique entre source et destination pour la durée de l'appel. La route est jalonnée de points de relais intermédiaires: en chacun de ces points, le message est mis en file d'attente jusqu'à ce qu'il puisse être envoyé vers le point de relais suivant ou la destination finale. De plus, chaque ligne peut multiplexer les messages pour une meilleure utilisation de sa capacité. Cette technique a des désavantages:

- la ligne de communication entre deux centres de commutation adjacents étant partagée par des messages ayant des sources et des destinations différentes, un message très long peut occuper la ligne pendant un temps considérable et donc retarder trop fortement des messages courts;
- une erreur détectée par le récepteur demande la retransmission du message entier, qui peut être très long.

Ces désavantages sont très atténués par la méthode de commutation de paquet. Les messages de l'utilisateur, en entrant dans le réseau, sont fragmentés en paquets de longueur maximum fixée et le transfert de l'information se fait paquet par paquet. L'avantage principal est que l'on stocke le paquet en mémoire centrale (du mini-ordinateur qui assure entre autres la commutation) alors que le message était généralement stocké en mémoire auxiliaire. La topologie d'un réseau à commutation de paquets est étudiée pour fournir au moins deux routes différentes entre chaque paire de noeuds. (Si une panne de ligne ou de noeud bloque une de ces routes, on pourra utiliser une autre route). Un problème de routage doit dès lors être résolu: les routes doivent être sélectionnées pour pouvoir répondre aux exigences du trafic et pour optimiser la performance du réseau généralement mesurée par le temps passé par le paquet ou le message dans le réseau.

Etablissons la structure générale du réseau.

Les ordinateurs des centres faisant partie du réseau, que nous appellerons Hosts, sont reliés chacun à un processeur d'interface de message, appelé Imp, qui est un mini-ordinateur. Les communications entre Hosts se font par l'intermédiaire de ces Imp, reliés



entre eux d'une certaine façon par des lignes full-duplex; l'ensemble des Imp et de ces lignes forment le sous-réseau de communication. Un Imp est donc aussi un interface de communication du sous-réseau. Il peut être connecté à plusieurs Hosts (et / ou terminaux). En plus d'un interface commun entre Host et Imp, un interface spécialisé est placé entre un Imp et un Host particulier.

Pour éviter une congestion de message du Host vers son Imp, la technique généralement employée est la suivante. Les messages sont transmis sur des chemins logiques entre Hosts, (Ces chemins n'ont pas de réalité physique). Le sous-réseau accepte seulement un message à la fois sur un chemin logique, d'autres messages ne pourront entrer sur ce chemin logique tant que le Imp source ne sera informé de l'arrivée au Host destination. Lorsque le chemin logique est débloqué, le sous-réseau prévient le Host source en lui envoyant un message spécial lui notifiant le chemin logique débloqué. Cette utilisation de chemin logique fournit une suite de messages dans le bon ordre. La communication entre Hosts est régie par un protocole "Host-Host" (ou "end-end") dont nous ne parlerons pas dans ce travail.

En général, les messages transmis par un Host vers son Imp ont une longueur maximum fixée (8064 bits dans le réseau Arpa). Si un message plus long doit être transmis, il sera scindé en blocs ne dépassant pas cette longueur maximum. Le Host émetteur attache un en-tête à chaque message (bloc) et envoie ce message à son Imp (avec les caractères "hardware" nécessaires à la transmission). Le Imp fragmente alors le message en paquets de longueur maximum fixée (1008 bits dans Arpa), forme un en-tête à attacher à chaque paquet en se servant de l'en-tête du message et d'informations supplémentaires pour l'utilisation du sous-réseau. L'Imp envoie alors les paquets (avec les caractères propres à la transmission) séparément dans le sous-réseau. Lorsque l'Imp relié au Host de destination a reçu tous les paquets d'un message, non nécessairement dans l'ordre, il les réassemble en un message en enlevant les en-têtes des paquets et en y attachant un en-tête identifiant le Host source et le chemin logique et délivre finalement ce message au Host destination. Il ne reste plus alors qu'à prévenir le Host source du déblocage du chemin logique. La technique de fragmentation en paquets est totalement invisible des Hosts.



La fonction d'interface entre un Host et le sous-réseau de communication comporte donc les tâches suivantes:

1. Fragmenter un long message en blocs de message de longueur maximum fixée.
2. Adapter le code des blocs de message à un format standard acceptable dans le réseau.
3. Attacher un en-tête avec information d'adresse et de contrôle à chaque bloc de message.
4. Attacher les caractères nécessaires à la transmission et l'information de contrôle d'erreur (CRC) à chaque bloc de message.
5. Conserver une copie des blocs de message non acquittés pour une retransmission possible.  
(Lorsqu'un récepteur reçoit un bloc de message, il renvoie à l'émetteur un message d'acquiescement lui notifiant que le paquet a été bien transmis et bien reçu (ACK) ou que la transmission n'a pas été correcte).
6. Réassembler les blocs de message reçus en messages.
7. Fragmenter les blocs de message longs en paquets de longueur maximum fixée.
8. Attacher un en-tête à chaque paquet.
9. Attacher les caractères nécessaires à la transmission à chaque paquet.
10. Réassembler les paquets reçus en blocs de message.
11. Conserver une copie des paquets non acquittés pour une retransmission possible.
12. Contrôler le taux d'arrivée pour éviter la congestion.

En général, les tâches 1 à 6 sont exécutées par le Host et les tâches 7 à 12 par l'Imp. (Une autre possibilité est de les exécuter toutes dans l'Imp; dans ce cas, il faut un processeur supplémentaire entre un Host et son Imp: lorsqu'un Imp reçoit un paquet, il ne doit pas savoir si le paquet provient d'un Host ou d'un Imp voisin).

Les communications dans le réseau se font par l'intermédiaire des Imp connectés aux différents Hosts (ou terminaux) et reliés entre-eux par des lignes full-duplex de grande capacité. Nous dirons que les Imp sont les noeuds du sous-réseau de communication, dans lequel le flux s'exprime en terme de paquets. Deux grandes philosophies sont employées à ce niveau dans les réseaux existants: le système "datagramme" et le système "circuit virtuel".



Dans le système "datagramme", chaque paquet de chaque message possède une information sur sa propre destination (qui fait partie du protocole). Chaque noeud le long de la route utilise cette information pour décider du prochain noeud auquel il faut envoyer le paquet. (D'autres informations telles que la disponibilité de charge peuvent être considérées.) De proche en proche, le paquet arrive finalement à destination.

Dans le système "circuit virtuel", chaque noeud fournit un système de voies logiques d'entrée et de sortie. La communication se fait en trois phases:

1. Etablir un circuit virtuel passant par une succession de noeuds connectant le noeud source au noeud destination.
2. Utiliser ce circuit virtuel pour tous les paquets se rapportant à un message (sauf si une panne se produit).
3. Déconnecter le circuit virtuel.

Dans ce dernier cas, le routage n'est exécuté qu'une seule fois pour une série de communications. Nous travaillerons par la suite dans un contexte "datagramme".

Un mécanisme simplifié de commutation de paquets entre deux Hosts est illustré à la figure 1.1. (RFNM, "request for next message", est le message qui débloque un chemin logique.)

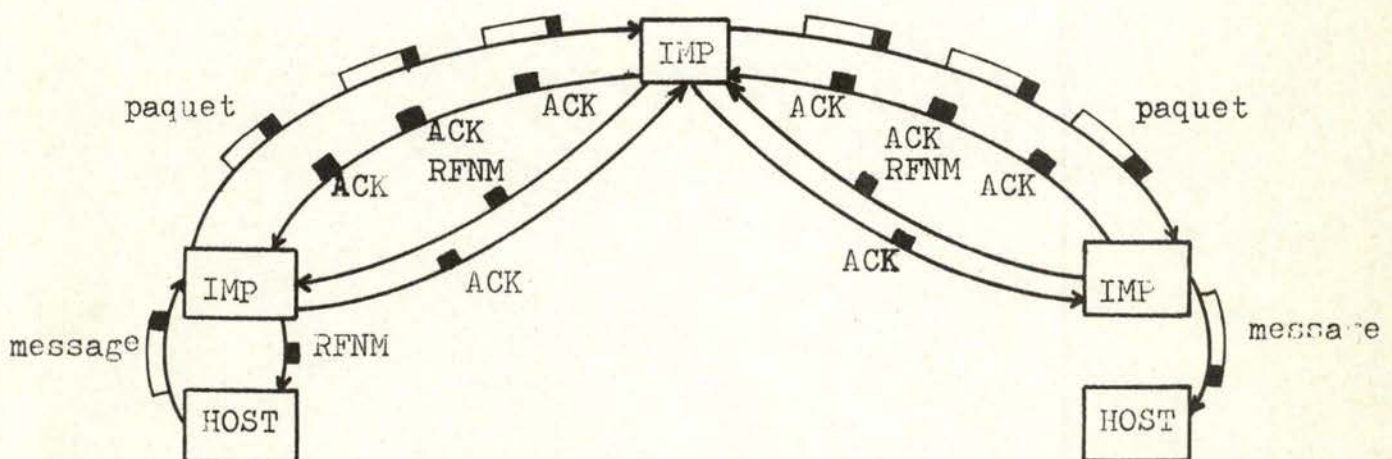


Fig. 1.1. commutation de paquets entre Hosts.

Le premier réseau à commutation de paquets est le réseau Arpa aux Etats-Unis. Il est opérationnel depuis 1969 et relie actuellement plus de 40 centres de recherche. Il s'agit d'un système à commutation de paquets distribué, où chaque noeud est un centre de commutation "store and forward", c'est-à-dire que les paquets sont dirigés de noeud en noeud jusqu'à leur destination, chaque noeud pouvant conserver dans une file d'attente une copie du paquet qui est envoyé au noeud voisin suivant, jusqu'à réception d'un acquit renvoyé par ce noeud voisin pour lui dire que le paquet a été bien reçu.

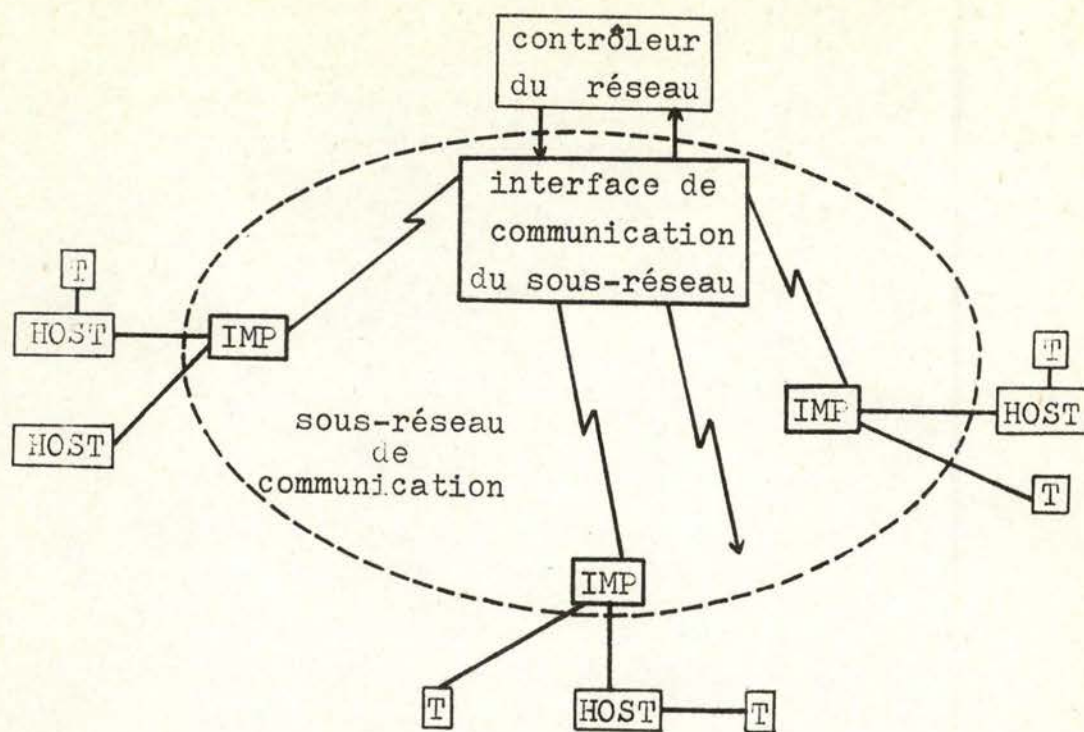
Un Imp possède des files d'attente de sortie pour chaque ligne vers les Hosts et les Imp qui lui sont connectés. Il contient un ensemble de routines pour traiter l'information entre les différents éléments:

- routine "Host-Imp"
- routine "Modem-Imp"
- routine "Imp-Modem"
- routine "Imp-Host"
- routine "tâche principale"
- routine "Time-Out"

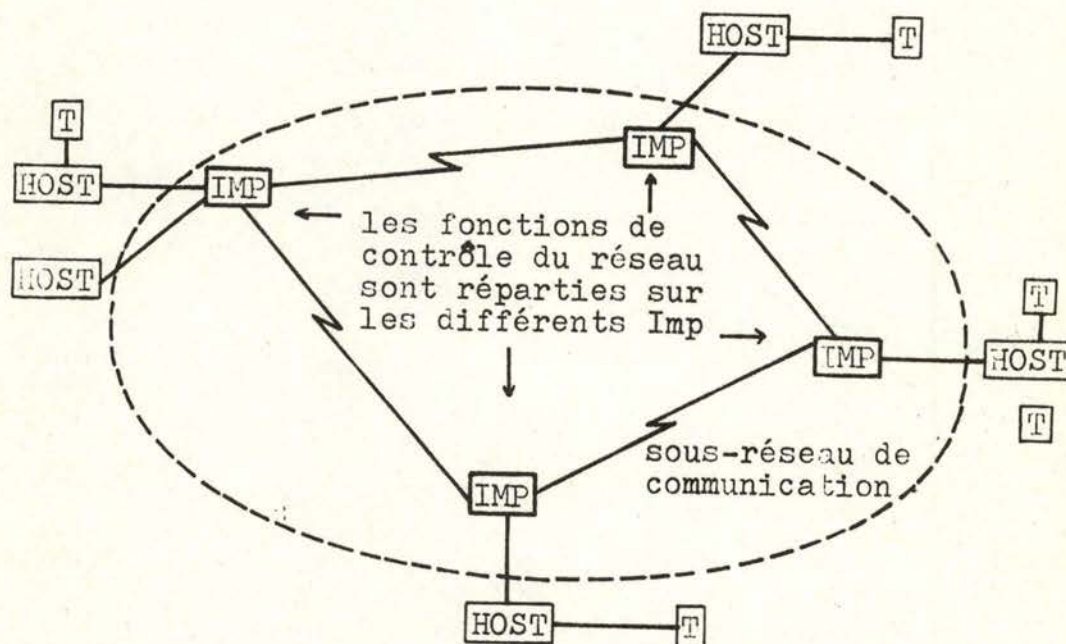
Nous détaillerons ailleurs les routines qui nous intéressent.

La figure 1.2. donne deux exemples de réseau.





a. Réseau centralisé



b. Réseau distribué

Fig. 1.2 Deux types de réseau.

## 1.2. Le sous-réseau de communication.

Le sous-réseau de communication est constitué de noeuds reliés entre-eux par des lignes full-duplex, en général de **grande** capacité. Chaque noeud comporte autant de modems que de lignes raccordées à ce noeud. La communication entre modem et processeur du noeud est régie par deux procédures: une procédure noeud-modem (NM) et une procédure modem-noeud (MN).

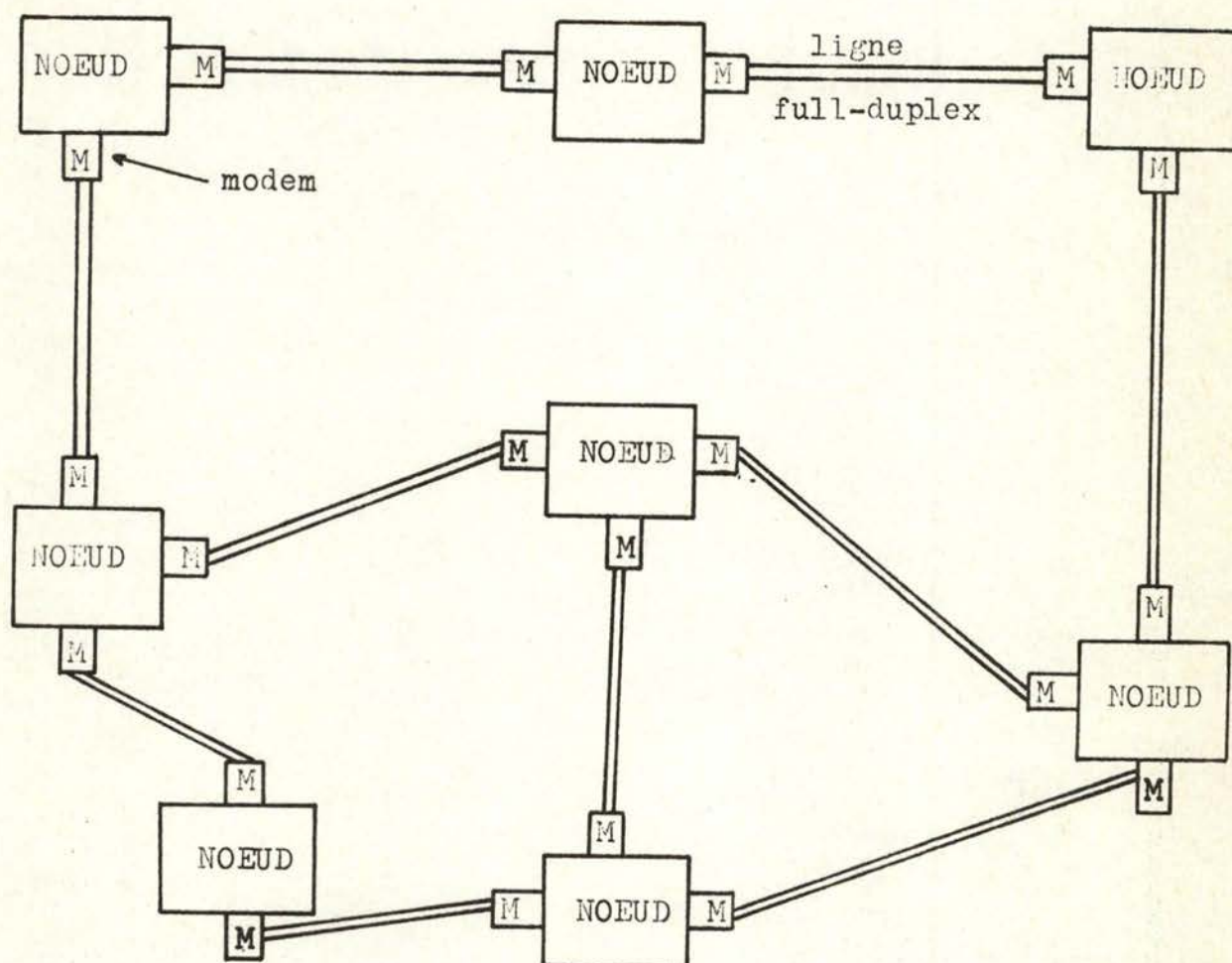


Fig. 1.3. Un sous-réseau de communication.

Nous allons décrire le protocole de base entre noeuds et la structure du noeud.



Si l'on veut utiliser au mieux la ligne entre deux noeuds, le protocole de base doit permettre l'émission de plusieurs paquets avant de recevoir le premier ACK. Les paquets envoyés et non acquittés sont conservés dans une file d'attente (SQ, sent queue) du noeud émetteur. Chaque ACK doit dès lors permettre une identification non ambiguë du paquet concerné, sans donner naissance à un overhead important. Pour cela, on divise chaque direction de la ligne full-duplex en  $N$  canaux logiques et on implémente sur chacun d'eux une procédure "stop and wait" avec une numérotation modulo-2 pour éviter les doublets résultant des ACK perdus et du mécanisme du time-out. (Lors de l'émission d'un paquet, on met en marche une horloge. Si l'ACK n'a pas été reçu lorsque l'horloge atteint une certaine heure, appelée time-out, on fera un nouvel essai de transmission du paquet; le nombre de retransmission est limité: si après ce nombre, le paquet n'est pas encore acquitté, on met en oeuvre une procédure spéciale (error recovery procedure).

Pour chaque paquet sortant, on indique le numéro du canal logique (3 bits si  $N = 8$ ) et le numéro modulo-2 du paquet sur ce canal logique (bit de parité) qui constitue l'identification locale du paquet.

Du côté émetteur d'un canal logique, le contexte est représenté par un bit d'état (SE) et le numéro du dernier paquet émis (NE). Du côté récepteur, on a seulement besoin du numéro du dernier paquet reçu (NR). Un paquet est accepté par le récepteur ssi il est correct et a le numéro attendu sur le canal utilisé. Comme ACK, le récepteur renvoie à l'émetteur l'identification locale ( $i, NR_i$ ) du dernier paquet reçu correctement, ce qui permet à l'émetteur de rendre libre un buffer (de la SQ) et le canal logique. Le diagramme d'état du protocole de base est donné dans l'annexe 1. Comme l'on a  $N$  canaux logiques avec une numérotation modulo-2, l'overhead associé aux ACK sera réduit si le récepteur renvoie à l'émetteur un bloc ACK constitué de  $N$  bits, chaque bit correspondant au n° du dernier paquet reçu correctement sur chacun des  $N$  canaux logiques. Ce bloc ACK peut être envoyé à l'émetteur de plusieurs façons: on peut soit réserver dans l'en-tête de chaque paquet une zone pour le bloc ACK du trafic inverse, soit utiliser un "paquet de contrôle" contenant seulement ce bloc ACK (appelé aussi paquet vide).



Les formats correspondants des paquets sont donnés sur la figure 1.4.

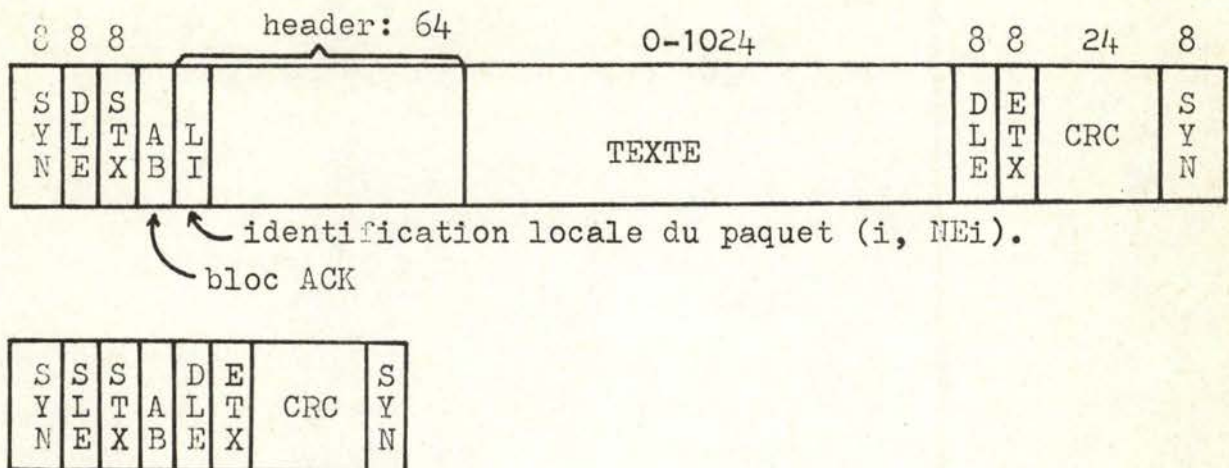


Fig. 1.4. Formats d'un paquet normal et d'un paquet de contrôle.

### 1.2.2. La structure du noeud.

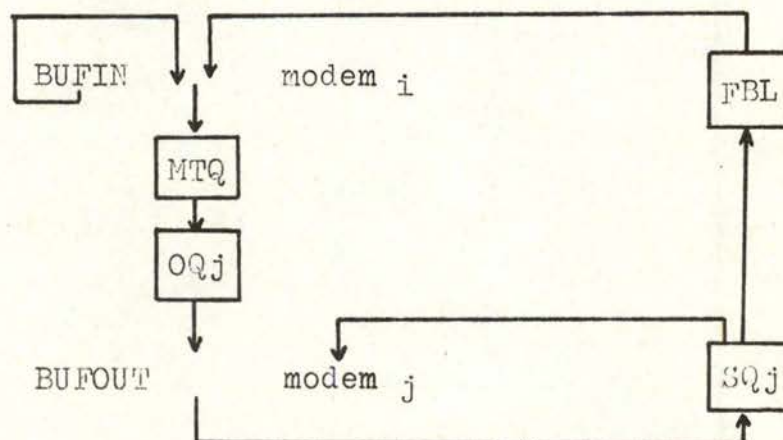


Fig. 1.5. Organisation des buffers.

L'organisation des buffers est donnée à la figure ci-dessus. Les buffers inutilisés sont groupés dans la liste des buffers libres (FBL). Un buffer est alloué au modem input à partir de cette FBL lorsqu'une opération DMA d'entrée est initialisée. A la fin du transfert, le buffer va dans la file d'attente de la tâche principale (MTQ). Puis il est attaché à la file d'attente de sortie (OQj) correspondant à la ligne de sortie sélectionnée avant d'être alloué au modem output lorsqu'une opération DMA de



sortie est initialisée. A la fin de l'émission, le buffer est attaché à la file d'attente des paquets envoyés ( $SQ_j$ ). Si le paquet doit être réémis, le buffer sera réalloué au modem output. Lorsque l'ACK correspondant aura été reçu, il retournera dans la FBL.

Les diverses routines exécutées dans un noeud sont décrites en annexe 1. Notons que la routine principale choisit sur base de l'en-tête et d'un algorithme de routage la ligne de sortie à utiliser et place le buffer contenant le paquet dans l'OQ correspondante. Pour la routine noeud-modem, plusieurs procédures peuvent être considérées suivant la politique employée pour la transmission du bloc ACK. Sa tâche est d'envoyer sur la ligne libre un nouveau paquet et de placer une copie du paquet dans la SQ. La routine modem-noeud s'occupe du contrôle de la transmission (examen du CRC) et du transfert du buffer contenant le paquet arrivé (si c'est un paquet régulier) dans la file d'attente de la tâche principale.



### 1.3 Comportement dynamique dans le sous-réseau

Dans le sous-réseau, la charge des différents trafics entre les noeuds "source" et les noeuds "destination" varie évidemment dans le temps.

Dans un noeud, les diverses files d'attente évoluent avec le trafic qui passe par le noeud.

Dans les réseaux à système " datagramme ", on emploie alors, en général, un algorithme de routage pour suivre au mieux l'évolution des différents trafics: un algorithme de routage adaptatif.

Nous parlerons dans le chapitre suivant des diverses méthodes de routage qui peuvent être utilisées dans les réseaux à commutation de paquets, et spécialement de l'une d'entre elles qui nous semble bien appropriée et dont des variantes ont été implémentées dans des réseaux existants ( Arpa, Tidas ). Nous verrons quelles sont les modifications à apporter à l'algorithme qu'elle utilise pour qu'il fournisse une mesure réelle de la charge moyenne sur une route, et pour qu'il se comporte de manière satisfaisante face à des pannes de ligne ou de noeud.

Un article de Mrs. Danthine et Eschenauer (3) met en évidence le comportement dynamique des files d'attente OQ et SQ d'un noeud, qui amène des phénomènes de saturation momentanée des huit canaux logiques d'une ligne de sortie du noeud. Nous étudierons ce comportement dynamique en relation avec l'utilisation de l'algorithme de routage adaptatif.

L'information utilisée par un tel algorithme doit être mise à jour à certains moments et le problème fondamental est de savoir comment et quand cette mise à jour devra être mise en oeuvre. Nous en discuterons au chapitre 3. Nous verrons que la mise à jour intervient dans un noeud à partir d' informations internes ( état de certaines files d'attente ) mais aussi à partir d'informations externes communiquées par les noeuds voisins par l'intermédiaire de messages de routage. Cet échange doit être réalisé pour refléter une situation toujours "actuelle" de l'état du sous-réseau.



Nous présenterons et nous analyserons au chapitre 4 des programmes de simulation développés pour étudier l'évolution des différentes files d'attente d'un noeud à travers le mécanisme d'échange entre deux noeuds.

Nous espérons pouvoir en tirer des conclusions quant à la manière de réaliser la mise à jour, ce qui permettrait d'atténuer ou d'éliminer des phénomènes propres à un routage adaptatif, tels que la formation de boucles et l'effet "ping-pong", définis au chapitre 3.

La nécessité de recourir à la simulation pour étudier le comportement d'un mécanisme d'échange entre noeuds provient de la complexité du système et du grand nombre de paramètres qui interviennent: on ne peut pas raisonnablement utiliser un modèle analytique représentant le système à étudier sans poser des hypothèses trop restrictives. De plus, la simulation permet d'observer les occurrences réelles des événements et donc le comportement macroscopique du système simulé, ce qui nous sera utile dans l'analyse des résultats.

LE ROUTAGE2.1. Le concept de routage.

Le but d'une procédure de routage est de choisir une route adéquate dans le réseau pour le transfert d'un paquet entre un émetteur initial (source) et un récepteur final (destination) sous des conditions variables dans le réseau. Il faut, en principe, que la procédure choisie puisse traiter le cas où, après une panne de ligne ou de noeud, le réseau puisse, après un certain temps, atteindre un état où l'échange du trafic entre les noeuds restants est possible. Cela demande au moins 2 chemins possibles entre chaque paire de noeuds. (Un exemple en est donné à la fig. 1.3.)

Il faut aussi qu'un pourcentage très élevé de données n'excède pas un âge maximum ou un temps de réponse donné lorsqu'elles atteignent les noeuds de destination. Cela demande une procédure efficace qui protège le réseau contre un phénomène de congestion et contre l'effet "ping-pong". (voir 3.1.)

Un bon routage doit, nous semble-t-il, :

- assurer une livraison rapide des messages;
- s'adapter aux changements de topologie du réseau, dus à des pannes de noeud ou de ligne;
- s'adapter aux variations de charge du trafic;
- ne pas diriger temporairement les paquets vers des noeuds saturés ou fortement chargés (au point de vue du contenu des files);
- être efficace contre l'effet "ping-pong";

On peut mentionner aussi la facilité d'installation et d'extension si l'on ajoute des lignes et/ou des noeuds.

Le routage peut s'effectuer de plusieurs manières très différentes que nous exposons brièvement ci-après.



## 2.2. Les méthodes de routage.

### 2.2.1. Classifications générale et fonctionnelle.

Les principales méthodes de routage utilisables dans les réseaux à commutation de paquets sont reprises dans la figure 2.1. Le problème de routage consiste à choisir une route optimale au sens d'un critère déterminé.

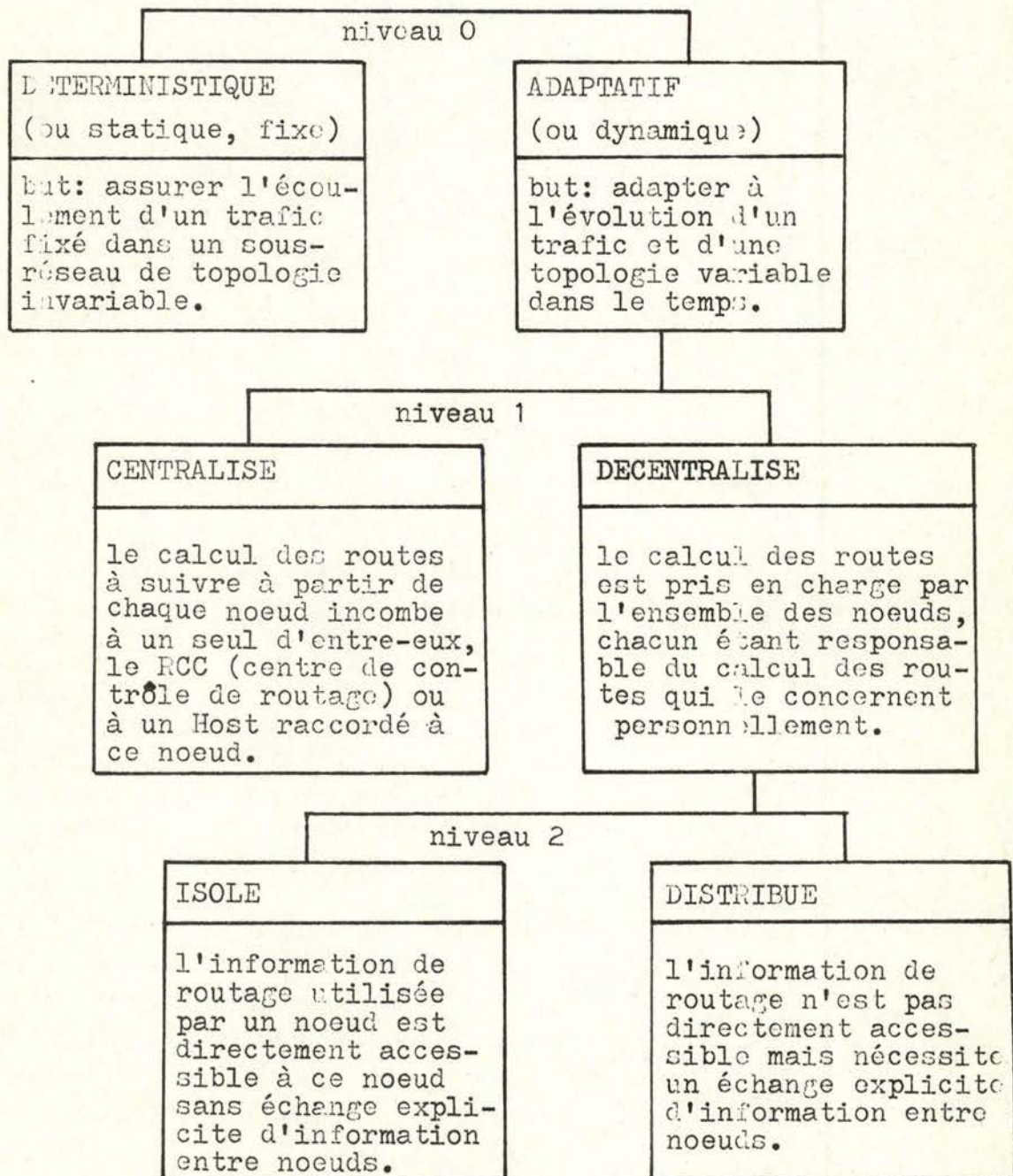


Fig. 2.1. les types de routage.



D'un point de vue fonctionnel, un algorithme de routage est un processus de prise de décision où les choix impliquent l'accès à une information sur l'état (information qu'il faudra mettre à jour) et se matérialisent par l'acheminement des paquets dans le sous-réseau.

### 2.2.2. Le processus de décision.

L'algorithme de décision doit prendre 2 décisions.

1. acheminer ou détruire le paquet traité (si le noeud de destination est inaccessible), c'est un problème d'accessibilité, le choix effectué est de nature binaire.
2. choisir les routes à suivre. C'est un problème de sélection des routes, qui est abordé dans une optique d'optimisation. Par la suite, nous choisirons comme fonction de coût à minimiser le délai, comme c'est le cas le plus souvent: des délais courts sont jugés indispensables pour un trafic interactif. Les délais sont additifs. On peut aussi envisager une minimisation du coût, si l'on veut utiliser de façon économique les ressources limitées que sont la capacité de la ligne, les capacités de traitement et de mémorisation des noeuds, ou bien une minimisation de la distance, ou encore une maximisation du "throughput" ou de la fiabilité.

Le processus de décision comporte donc 2 algorithmes dont les rôles sont précisés à la figure 2.2.

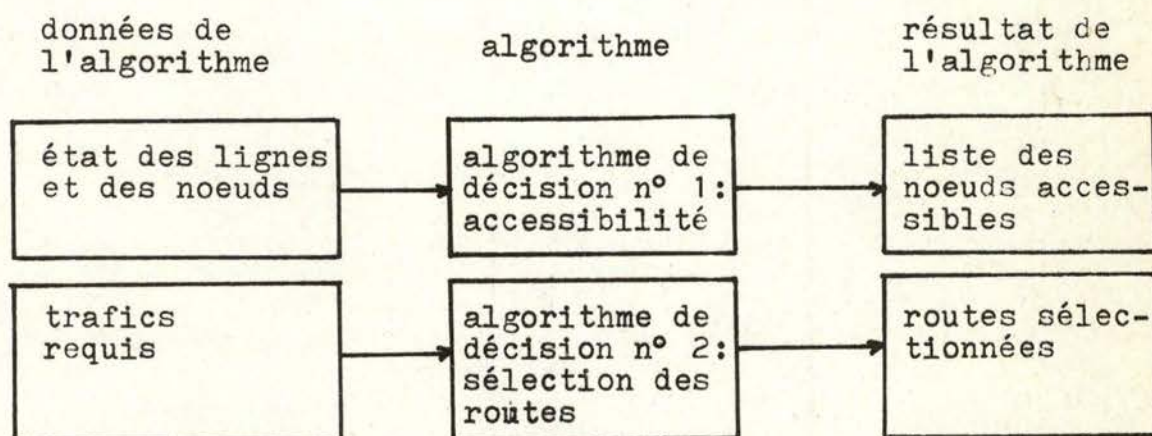


Fig. 2.2.



Le processus de mise à jour fournit aux algorithmes de décision l'information mise à jour sur l'état du réseau qui leur est nécessaire. Il ne concerne que les routages adaptatifs.

Dans le routage isolé, la mise à jour est le fruit d'observations locales; dans les routages distribué et centralisé, elle utilise en plus des observations locales, les observations distantes fournies par les messages de routage échangés entre noeuds. Le contenu de ces messages de routage dépend des spécifications de l'algorithme de décision et spécialement du choix de la fonction de coût (le délai). Quant à leur propagation, elle peut-être périodique, avec ou sans synchronisation entre noeuds, ou aperiodique: l'émission d'un message de routage étant déclenché par exemple par une variation de grandeur mesurée supérieure à un certain seuil.

#### 2.2.4. Le processus d'acheminement.

On exécute les décisions prises au niveau du processus de décision. On choisit une approche qui consiste à laisser l'initiative de l'acheminement successivement à chacun des noeuds intermédiaires. En pratique l'acheminement à un noeud  $p$  s'effectue en consultant une table de routage  $TR^p$  fournie par le processus de décision. Le nombre d'entrées de cette table, dans le cas général, est trois: l'élément  $TR^p(i,j,k)$  donne l'identité de la  $k$ ième ligne de sortie attribuée par le noeud  $p$  aux paquets dont  $i$  et  $j$  sont respectivement le noeud source et le noeud destination. Les indices  $i,j,k$ , varient de 1 à  $i_{\max}$ , 1 à  $j_{\max}$ , 1 à  $k_{\max}$  respectivement ( $k_{\max}$  est le nombre de choix distincts d'une ligne de sortie).

- Si  $i_{\max} = 1$ , le routage est indépendant de la source.
- Si  $j_{\max} = 1$ , le routage est indépendant de la destination.
- Si  $k_{\max} = 1$ , tous les paquets relatifs à une paire source destination  $(i,j)$  sont émis sur une seule et même ligne de sortie.

Les dimensions  $i_{\max}$ ,  $j_{\max}$ ,  $k_{\max}$  de la table de routage  $TR$  constituent un aspect essentiel de l'évaluation du coût d'un algorithme de routage, car elles conditionnent à la fois l'occupation mémoire et la charge de calcul des routes.



## 2.2.5. Caractéristiques des principaux types de routage.

### 2.2.5.1. Les routages déterministiques.

Parmi les différents types de routage déterministique, certains ont des performances trop médiocres:

- le "flooding" ou le "flooding sélectif": chaque noeud recevant un paquet l'émet systématiquement sur toutes ses lignes de sortie ou sur un nombre limité d'entre-elles respectivement, à moins que le paquet ne soit arrivé à destination ou n'aie déjà été émis par ce noeud.
- le routage aléatoire: le choix de noeud suivant s'effectue en respectant une distribution de probabilité sur les noeuds voisins, fixée une fois pour toutes.

Les autres types sont:

- le routage fixe ou non bifurqué: il n'offre qu'une ligne de sortie par paire source-destination  $(i, j)$ . Il est élémentaire si  $i_{\max} = 1$ , c-a-d s'il est indépendant de la source.
- le routage bifurqué: il offre plusieurs lignes de sortie aux paquets d'une paire de source-destination  $(i, j)$ .

Ces deux méthodes donnent une bonne idée des performances que l'on peut attendre d'un réseau existant et constitue un outil mathématique de première importance en ce qui concerne la synthèse d'un réseau non existant. Elles sont par contre totalement inadaptées aux exigences d'une implémentation réelle, domaine réservé des routages adaptatifs.

### 2.2.5.2. Les routages adaptatifs.

- a) Le routage centralisé: le processus de décision est centré au niveau du RCC, les processus de mise à jour et d'acheminement restent distribués au niveau de l'ensemble des noeuds. Il en résulte un double flux d'information comprenant d'une part les données mises à jour sur l'état du réseau (émises par les noeuds vers la RCC) et de l'autre les tables de routage (émises par le RCC vers les autres noeuds).



Il présente les faiblesses suivantes: tout le système s'effondre si le RCC tombe en panne: il faut donc, par exemple, doubler le RCC; de plus, des pannes de noeuds ou de lignes peuvent isoler une région du réseau démunie de RCC; enfin, la densité du trafic associée aux messages de routage est inégalement répartie dans le réseau: forte au voisinage du RCC, elle est faible à la périphérie, ce qui entraîne une charge utile faible des lignes au voisinage du RCC. La figure 1.2.a. donne un exemple de réseau centralisé où l'on peut utiliser un routage centralisé.

b) Le routage décentralisé isolé:

- le routage "hot potato": on y dirige tout paquet arrivant dans un noeud sur la ligne de sortie dont la file d'attente est la plus courte, quelles que soient la source et la destination du paquet. Il en résulte de très fréquentes formations de boucles sur le parcours desquelles les paquets tournent en rond (due à l'indépendance du critère de la file d'attente la plus courte vis-à-vis de la destination), l'allongement considérable du trajet moyen en nombre de noeuds visités, ainsi que les délais moyens très élevés et des throughputs très faibles.
- le routage par "backwards learning" est indépendant de la source et dépendant de la destination. Un échange implicite d'information entre noeuds se fait par l'intermédiaire du "hand-over number" contenu dans l'entête des paquets: ce nombre est incrémenté à chaque passage du paquet en un noeud de façon à obtenir une estimation du délai subi par le paquet lors de son transfert dans le sous-réseau. Ce routage nécessite l'utilisation des notions "feedbacks positifs et négatifs" car dans un routage isolé, un noeud ne peut-être tenu au courant de la panne d'un autre noeud ou d'une ligne de sortie distincte de ses propres lignes de sortie. Si la meilleure route est bloquée par une panne, le feedback positif est une démarche



nécessaire d'apprentissage destinée à découvrir de meilleures routes, mais il serait souhaitable d'éviter à priori les essais conduisant à de très mauvaises routes; c'est le rôle du feedback négatif.

La recherche des meilleures routes se traduit par des oscillations dans leur choix, ce qui fait que ce routage est fondamentalement instable, même en régime établi.

Ces différentes méthodes de routage sont décrites plus complètement par E. Bauwens dans (1).

### c) le routage décentralisé distribué

Nous nous intéresserons par la suite à un routage de ce type. Aussi décrivons-nous au paragraphe suivant un routage de ce type/ le routage S.Q. + Bias + P.U.

(Shortest Queue + Bias + Périodic Updating) c-a-d- un routage structuré en trois parties correspondant à la file de sortie la plus courte, à un biais et à une mise à jour périodique).

Un routage de ce type a été implémenté dans plusieurs réseaux (Arpa aux E.U., Tidas en Suède notamment).

## 2.3. Le routage S.Q. + Bias + P.U.

### 2.3.1. Description

#### a) Définition

Le routage S.Q. + Bias + P.U. est un routage adaptatif et distribué (utilisant donc un échange explicite d'information entre noeuds), dépendant de la destination et indépendant de la source.

Le critère de sélection des routes est, comme on l'a déjà dit, le délai minimum. Chaque noeud p possède une table des délais  $TD^p$  à deux entrées correspondant à la destination et à la ligne de sortie. L'application de l'algorithme de sélection fournit une table de routage  $TR^p$  donnant pour chaque destination la ligne de sortie optimale.



La structure d'estimation des délais est donnée par la formule suivante:

$$TD^p(j,k) = W^p(k) + D^p + VDM_{rk}^p(j) \quad (2.1)$$

où:

- $TD^p(j,k)$  est l'élément de la table des délais du noeud  $p$  qui correspond au noeud de destination  $j$  et à la ligne de sortie  $k$ ; c'est le délai estimé d'un transfert de  $p$  vers  $j$  via la ligne de sortie  $k$ ;
  - $W^p(k)$  est l'estimation du délai d'accès à la ligne de sortie  $k$  du noeud  $p$ , c-a-d du temps passé dans la file d'attente de la ligne  $k$ ;
  - $D^p$  est le biais (considéré par Fultz (5) comme un paramètre à ajuster par simulation pour obtenir un délai moyen minimum). Le terme  $W^p(k) + D^p$  représente la partie de la ligne de sortie  $k$  dans le délai total et constitue une information directement accessible au noeud  $p$ .
- Par contre, le terme  $VDM_{rk}^p(j)$  n'est connu du noeud  $p$  que grâce à l'échange d'information entre noeuds voisins. C'est le  $j^{\text{ème}}$  élément du vecteur des délais minima reçu au noeud  $p$  en provenance du noeud voisin de  $p$  qui se trouve à l'autre extrémité de la ligne de sortie  $k$ . Les vecteurs VDM sont transmis sous forme de paquets de routage.

#### b) la table des délais

Cette table sert de donnée à l'algorithme de sélection des routes. Ses éléments sont mis à jour périodiquement.

ligne de sortie destination	1...	[k]	...NLS(p)
1	$TD^p(1,1)...$		$...TD^p(1,NLS(p))$
⋮	⋮		⋮
j		$TD^p(j,k)$	
⋮	⋮		⋮
N	$TD^p(N,1)...$		$...TD^p(N,NLS(p))$

Fig. 2.3. Table des délais  $TD^p$



c) Sélection de la ligne de sortie optimale.

La meilleure ligne de sortie  $k_{opt}(j)$  est calculée en utilisant le critère du délai minimum:

$$TD^P(j, k=k_{opt}(j)) = \min_{k=1, \dots, NLS(p)} TD^P(j, k) \quad (2.2)$$

$j=1, \dots, N.$

Si dans la table de la fig. 2.3. la ligne  $k$  a été sélectionnée pour la destination  $j$ , on aura la table de routage suivante:

destination	ligne de sortie optimale
1	$k$
.	
.	
$j$	
.	
.	
$N$	

$TD^P \xrightarrow{(2.2)}$

Fig. 2.4. table de routage  $TR^P$

d) Le vecteur de délais minima  $VDM^P$

Chaque noeud calcule un vecteur des délais minima qu'il transmet à tous ses voisins. Le vecteur des délais minima calculé au noeud  $p$  est défini comme suit:

$$VDM^P(j) = TD^P(j, k_{opt}(j)) \quad j \neq p, j=1, \dots, N \quad (2.3.)$$

$$VDM^P(p) = 0$$

où  $k_{opt}$  vérifie (2.2).

e) Le processus de mise à jour.

-mise à jour interne:

il faut tenir compte des modifications de  $W^P(k)$ ; cette mise à jour est déclenchée par l'arrivée d'un paquet en file d'attente (incrémentement) ou par l'émission d'un paquet sur la ligne de sortie (décrémentement).



- mise à jour externe:

il faut ici tenir compte des variations de la grandeur extérieure  $VDM_{rk}^p(j)$ ; cette mise à jour est déclenchée par l'arrivée d'un vecteur VDM provenant d'un noeud voisin. Le calcul et l'émission du VDM d'un noeud s'effectue de façon périodique (toutes les  $T_u$  secondes,  $T_u = 2/3$  dans ARPA).

En pratique, les instants d'émission par les différents noeuds de leur VDM ne seront pas synchrones.

La route à suivre peut:

- soit être recalculée pour chaque paquet
- soit être calculée  $n$  fois par période  $T_u$  et mémorisée dans la table TR pendant  $T_u/n$  sec.

f) L'algorithme d'accessibilité.

Chaque noeud  $p$  émet périodiquement son "minimum-hop-vector"  $MHV^p$  vers tous les noeuds voisins, le vecteur  $MHV^p$  est défini par:

$$MHV^p(j) = \text{nombre minimum de "hops" entre } p \text{ et } j, \\ j \neq p, j=1, \dots, N$$

$$MHV^p(p) = 0$$

Chaque fois que le noeud  $p$  reçoit un MHV issu d'un noeud voisin, il recalcule son propre MHV selon:

$$MHV^p(j) = 1 + \min_{k:k=1, \dots, NLS(p)} MHV_{rk}^p(j), \\ j \neq p, j=1, \dots, N$$

$$MHV^p(p) = 0 \quad (2.4.)$$

Où  $MHV_{rk}^p$  est le MHV reçu en  $p$  en provenant du noeud qui se trouve à l'autre bout de la ligne de sortie  $k$ .

Cet algorithme est semblable à celui du routage S.Q.+ Bias + P.U. à ceci près que le délai a été remplacé par la distance évaluée en "hops" (c-a-d en nombre de noeuds).

Lorsque le noeud  $j$  devient inaccessible pour le noeud  $p$ , la distance minimale de  $p$  à  $j$  en "hops" se met à diverger. Dès lors, lorsque  $MHV^p(j)$  dépassera une constante à fixer, le noeud  $j$  pourra être considéré comme inaccessible à partir de  $p$ .



2.3.2. Remarque.

Nous avons vu que la table des délais  $TD^p$  contient les délais estimés d'un transfert de  $p$  vers les autres noeuds via les différentes lignes de sortie de  $p$ . Ceci s'explique de la façon suivante.

Supposons qu'un message doive être envoyé du noeud  $p$  au noeud  $j$ . La route doit alors passer par un nombre de noeuds  $n_1, n_2, \dots, n_l$ . Soit  $T_{qr}(t)$  le délai total entre les noeuds  $q$  et  $r$  au temps  $t$ , incluant les temps d'attente dans les files et les temps de transmission. Le message est au noeud  $p$  et doit choisir la meilleure route pour finalement atteindre le noeud  $j$ . Le délai total du message peut s'écrire:

$$\begin{aligned} T_{pj}(t) = & T_{pn_1}(t) \\ & + T_{n_1n_2}(t + T_{pn_1}(t)) \\ & + T_{n_2n_3}(t + T_{pn_1}(t) + T_{n_1n_2}(t + T_{pn_1}(t))) \\ & + \dots \end{aligned}$$

Quand la décision doit être prise en  $p$ , seul le premier terme de l'expression ci-dessus est connu. Donc une comparaison exacte entre les différentes routes n'est pas possible. Une solution approximée doit être utilisée: la plus évidente est de remplacer le  $T_{qr}(.)$  par une estimée  $\hat{T}_{qr}$ .

Mais même dans ce cas, il n'est pas réaliste du tout de résoudre un problème de prédiction si complexe en détail pour chaque message circulant dans le réseau. Cependant, si le réseau peut être considéré comme stationnaire, on peut utiliser une forme simplifiée de prédiction. Lors du routage, chaque noeud utilise les valeurs les plus récentes des délais estimés dont il dispose et envoie le message sur la première ligne de la route ayant le délai minimum pour atteindre le noeud destination. On établit donc pas la route absolue entre source et destination mais seulement celle qui doit mener au noeud suivant vers la destination finale.

Nous reviendrons au chapitre suivant sur l'estimation du délai d'attente dans les files de sortie et sur l'estimation du délai



pour atteindre le noeud de destination à partir du noeud suivant. La précision de ces estimations dépendra de la façon de mesurer l'attente en file et de la façon d'échanger de l'information de routage entre noeuds voisins.



### CHAPITRE III

#### CONSIDERATIONS

#### SUR LA DYNAMIQUE DES MECANISMES D'ECHANGE

Dans ce chapitre, nous parlerons du routage S.Q. + Bias + P.U. tel qu'il est défini au chapitre précédent, des problèmes posés par son utilisation, des modifications que l'on peut y apporter en relation avec le problème fondamental qui se pose lorsqu'on utilise un routage adaptatif: la formation de boucles; mais aussi de modifications plus fondamentales qui pourraient amener à un algorithme qui produise une mesure de la charge réelle moyenne sur une route, par une estimation plus précise des délais. Nous verrons aussi que l'algorithme d'accessibilité peut se baser sur des termes propres à l'algorithme général de routage, plutôt que sur des vecteurs spécifiques transmis sous forme de paquets de routage (vecteurs MHV), ce qui permettra de se passer de ces paquets et donc d'obtenir une plus grande capacité utile des lignes entre noeuds.



### 3.1. Considérations sur un routage adaptatif S.Q. + Bias + P.U.

La structure de ce routage adaptatif est définie au chapitre 2. Lorsqu'un noeud reçoit les vecteurs VDM de ses voisins (c-a-d les tables de routage de ses voisins), il construit une seule table de routage donnant, pour chaque destination, la ligne de sortie optimale. Pendant toute la période  $T_u$  de  $2/3$  de secondes, c'est-à-dire jusqu'à l'échantillonnage suivant, il n'y a aucune modification de cette table, quelle que soit l'évolution des files d'attente du noeud. Ce n'est qu'au moment précédent la transmission de sa propre situation à ses voisins que le noeud considéré fait un échantillonnage instantané sur les longueurs (en nombre de paquets) de ses files d'attente. Cette façon de procéder, invariante vis à vis des conditions de charge dans le noeud et du trafic pendant la période  $T_u$  semble à tout le moins discutable.

Lorsqu'une route passe par un grand nombre de noeuds, même si l'échantillonnage sur la longueur des files se fait instantanément et comporte une grande variation par rapport à la valeur moyenne réelle de la file, on pourrait, d'après les concepteurs d'Arpa, compter sur une compensation partielle, lorsque, par le jeu des associations de tables de routage des différents noeuds, on fait l'addition des différents éléments. Cette explication me semble pas évidente et le problème ne serait pas résolu pour autant lorsque le nombre de noeuds est relativement petit. Dès lors, une politique supplémentaire consiste, lorsqu'on a reçu de nouvelles informations à un noeud et que l'on met donc à jour la table de routage de ce noeud, à comparer la ligne de sortie optimale obtenue pour chaque destination à celle qui était optimale précédemment.

Par la technique du "hold-down", on persiste à maintenir pendant un certain temps ( $3 T_u = 2$  secondes) l'ancienne ligne de sortie optimale lorsque la situation se dégrade; on essaie de ne pas laisser influencer dynamiquement par une légère dégradation des résultats de l'optimisation théorique.

Il faut donc que la situation se dégrade fortement sur une ligne pour que l'on change le routage.



Cette technique va quelque peu à l'encontre du but fixé par l'emploi d'un routage adaptatif, l'adaptation à la charge du réseau, tout d'ailleurs comme le terme de "Bias" volontairement fixé assez important.

En ce qui concerne le terme "Shortest Queue", il faudrait introduire une valeur représentative de la charge moyenne de la ligne, l'algorithme de routage fournirait une mesure réelle de la charge moyenne sur une route. Pour cela, il nous semble que l'on ne peut pas se limiter au nombre de paquets dans les files de sortie, mais qu'il faudra d'une manière ou d'une autre tenir compte de la longueur des paquets. Il serait intéressant pour cela de dégager une fréquence d'échantillonnage dans le noeud, en tenant compte des divers types de paquets et de leur longueur, des conditions de charge du réseau, et à partir de celle-ci déterminer une période moyenne d'échantillonnage. L'échantillonnage obtenu devra respecter le théorème de Shannon.

En ce qui concerne le nombre maximum de buffers par ligne, on sait qu'il y a un maximum de 8 buffers dans la S.Q. (correspondant aux huit canaux logiques d'une ligne). Dans le réseau Arpa, le nombre maximum de buffers dans (OQ+SQ) est huit. Ce qui revient à dire que si un paquet arrive, après le routage, devant une ligne dont les huit canaux logiques sont occupés, il ne pourra être maintenu dans le noeud (en OQ). Dans ce cas, on jette le paquet. Ce paquet était néanmoins toujours présent dans le noeud précédent car on le jette avant d'avoir renvoyé l'ACK à ce noeud précédent. On voit que le temps de retour de l'ACK sera alors très élevé et que le délai total du paquet dans le sous-réseau en sera évidemment affecté.

Plus la charge du réseau augmentera, plus on jettera des , et plus le délai moyen total d'un paquet augmentera.

Une politique de transmission adéquate des ACK peut atténuer fortement cet inconvénient.

On peut noter qu'Arpa utilise actuellement pour la transmission des blocs d'ACK le protocole B (cf annexe 1), après avoir abandonné le protocole A en 1972.



L'utilisation du protocole C, qui utilise uniquement des paquets de contrôle pour la transmission des ACK, permet de libérer les buffers de la SQ plus tôt sans augmenter les délais puisque la transmission des ACK sur une ligne de sortie est prioritaire par rapport à la transmission des paquets normaux. Aussi utilisons-nous dans les divers essais que nous réalisons par simulation, une politique d'acquittement suivant le protocole C.



### 3.2. La formation de boucles - l'effet "ping-pong".

Tous les routages adaptatifs sont concernés par la formation de boucles.

Intéressons-nous à la topologie de la figure 3.1. et au routage des paquets de C vers A, pour un routage distribué avec mise à jour périodique et synchrone (c'est-à-dire que l'émission des messages de routage par les différents noeuds s'effectue simultanément toutes les  $T_u$  secondes.)

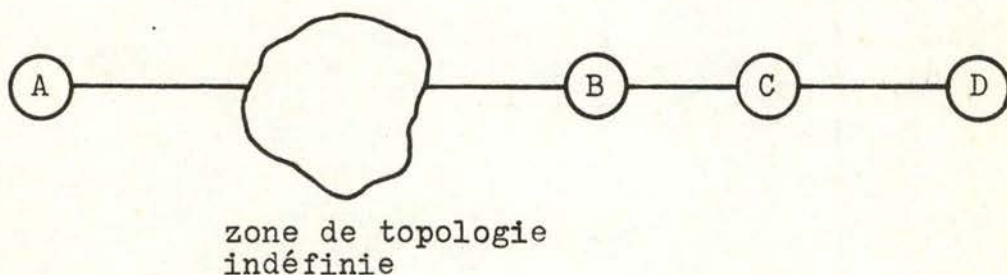


Fig. 3.1.

Une augmentation de délai importante, due par exemple à un accroissement de trafic  $B \rightarrow A$  entre les instants  $nT_u$  et  $(n+1)T_u$ , donne lieu à un effet "ping-pong"  $C \rightleftharpoons D$  pendant un certain temps: les paquets destinés à A à partir de C passeront par D car la ligne de sortie optimale de C vers la destination A est la ligne  $C \rightarrow D$ ; en D, les paquets seront dirigés vers C car la ligne de sortie optimale de D vers la destination A est la ligne  $D \rightarrow C$ .

Cette apparition d'effet ping-pong provient de ce que l'information sur l'accroissement de trafic entre B et A n'est pas simultanément en la possession de tous les noeuds entourant C (c'est-à-dire B et D), l'un d'entre-eux (D) étant plus éloigné que l'autre (B) de la zone B-A où se produit la modification de trafic. Cette information sera en la possession de B en  $(n+1)T_u$  et de D en  $(n+3)T_u$  puisque il y a un noeud entre B et D et que l'information ne se propage que de noeud en noeud toutes les  $T_u$  secondes.

Or la simultanéité d'acquisition en B et D de l'information sur la modification d'état du réseau dans la zone B-A est vitale puisque chacune des deux routes (la route directe  $C \rightarrow B \rightarrow \dots \rightarrow A$



et la route indirecte  $C \rightarrow D \rightarrow C \rightarrow B \rightarrow \dots \rightarrow A$ ) passe par la zone B-A.

Plusieurs moyens d'éviter un tel phénomène ont été proposés mais aucun ne semble totalement efficace.

On peut citer par exemple:

- l'adoption d'un terme de "Bias"  $D^P$  très élevée (cf (5)), de façon à ce que l'accroissement de délai reste inférieur à  $2 D^P$ . Cela présente un inconvénient important: dans la formule générale du délai (2.1.), si on prend le cas limite  $D_p \rightarrow \infty$ , il n'y a plus d'adaptation à la charge du réseau et on est dans le cas d'un routage fixe. On va donc alors à l'encontre d'un des buts fondamentaux de routage adaptatif.
- la méthode du "hold-down", déjà citée, dont le principe est le suivant: lorsque le délai de la meilleure route se dégrade, au point qu'elle cesse de l'être, devenant ainsi "l'ex-meilleure route", on continue cependant:

- à utiliser l'ex-meilleure route pour l'acheminement des paquets;
- à utiliser le délai dégradé de l'ex-meilleure route (et non celui de la meilleure route) pour la mise à jour des délais aux noeuds voisins, et cela pendant un certain temps.

Le but poursuivi est d'attendre que l'ensemble des noeuds voisins du noeud considéré aient eu connaissance de la modification d'état du réseau à laquelle correspond la dégradation de performance détectée.

(Dans Arpa, la durée du "hold-down" a été fixée à 2 secondes.)

Cette technique, bien qu'efficace dans certaines situations, est catastrophique dans d'autres, où elle prolonge la durée de vie naturelle des boucles. Nous en donnons un exemple, tiré de Naylor (10): reprenons la topologie de la fig. 3.1. et intéressons nous aux délais estimés des noeuds C et D vers le noeud éloigné A.



Les délais estimés dans un paquet de routage arrivant à un noeud sont comparés avec les entrées de la table de routage du noeud plus le délai du noeud vers son voisin (en général 4 unités + 1 par paquet dans la file de sortie).

La plus petite de ces deux valeurs remplace l'estimée de la table de routage. Si l'estimation des délais augmente de plus de 8 unités sur une ligne donnée, le noeud entre en "hold-down".

Soit:  $d/B$ : le délai estimé vers le noeud A lorsque B est le voisin suivant sur le chemin vers A.

$B!$  : hold-down sur la ligne vers B.

B C: le noeud B envoie une mise à jour de routage vers C.

Voyons ce qui se passe lorsque le délai estimé de C vers A augmente graduellement, en supposant qu'il y a deux fois plus de paquets de routage de D vers C que de C vers D.

	<u>Noeud C</u>	<u>Noeud D</u>
$C \rightarrow D$	$d/B$	$d+4/C$
$B \rightarrow C$	$d+5/B$	
$D \rightarrow C$		
$B \rightarrow C$	$d+9/B$	
$D \rightarrow C$	$d+8/D$	
$C \rightarrow D$		$d+13/C!$
$D \rightarrow C$	$d+17/D!$	

Une boucle apparaît puisque finalement le noeud C entre en hold-down en  $d+17$  alors que le noeud suivant vers la destination A est le noeud D (on continue à utiliser l'ex-meilleure route qui était celle qui passe par D en  $d+8$ ).

Nous verrons au paragraphe 3.4. que, en calculant d'une certaine façon les VDM, on peut réduire la probabilité de formation de boucles.



### 3.3. La mise à jour.

Le problème central dans un algorithme de routage adaptatif distribué est celui de la mise à jour de la table de routage. Quand et comment faut-il l'effectuer?. Nous avons vu que l'on peut distinguer une mise à jour interne, due aux variations dans les files d'attente du noeud lui-même et une mise à jour externe qui intervient lorsque des noeuds voisins se communiquent leur situation actuelle.

Nous proposons dans les paragraphes suivants divers moyens de réaliser ces mises à jour; en tenant compte notamment de l'accessibilité à un noeud et des pannes de lignes.

### 3.4. La mise à jour interne.

Cette mise à jour porte sur le terme  $W^p$ , c'est-à-dire sur l'estimation du délai d'accès à chacune des lignes de sortie du noeud  $p$ . En général, lorsque cette mise à jour a été effectuée, on n'a pas tenu compte de la longueur des paquets mais seulement du nombre de buffers (ou de paquets) dans chaque file  $OQ$  du noeud ou dans chaque ensemble ( $OQ + SQ$ ). Cependant, si une des files contient deux paquets très courts et si une autre file contient un paquet très long et si l'estimation du reste du délai total vers la destination finale est la même dans les deux cas, on choisissait plutôt la ligne de sortie de la première file bien qu'elle contienne plus de paquets.

Pour refléter au mieux la charge réelle d'une ligne, on peut travailler de la façon suivante:

- chaque fois qu'un paquet est dirigé vers une ligne de sortie déterminée (donc après que l'algorithme de routage lui aie été appliqué), on incrémente tous les éléments de la colonne correspondante de la table des délais du noeud d'un nombre proportionnel à la longueur du paquet. Le nombre le plus judicieux à choisir est la durée d'émission du paquet, exprimée en msec (qui s'obtient en divisant la longueur du paquet (en bits) par la capacité de la ligne (en Kbits/sec ou bits/msec)), puisque la somme des durées



d'émission des paquets en OQ donne une estimation précise du délai d'accès au noeud voisin sur la ligne.

- chaque fois qu'un paquet a été entièrement transmis, on décrémente les éléments de la même colonne de la TD de ce même nombre.

En procédant de cette façon, on ne parle pas de la mise à jour de la table de routage, seule la table des délais suit les modifications internes. Nous verrons plus loin comment la table de routage peut être modifiée uniquement en fonction du terme "Shortest Queue".

Quant à la file SQ, on ne peut pas tenir compte de la longueur des paquets qui s'y trouvent puisque la durée en SQ d'un paquet ne dépend pas de sa longueur (elle dépend essentiellement du trafic inverse entre les 2 noeuds et de la façon dont l'ACK est renvoyé).

En employant le protocole C pour la retransmission de l'ACK (cf annexe 1), on sait que l'on libère au plus tôt les canaux logiques d'une ligne. Si malgré cela, on atteint un état où les huit canaux logiques sont occupés, on peut toujours placer les paquets dans l'OQ: ils y resteront plus longtemps; à ce moment là, le délai estimé d'attente devant une ligne de sortie augmentera puisque rien ne sera transmis, et une autre route sera peut-être choisie pour acheminer les paquets suivants. Il faut évidemment pour cela que le nombre de paquets en (OQ + SQ) ne soit pas limité à 8.

On peut faire intervenir à ce niveau le fait qu'une ligne devienne non opérationnelle, ce qui est déclaré après un certain nombre d'essais de transmission infructueux. Lorsque cela se produit, les éléments de la colonne correspondante de la table des délais sont incrémentés d'un nombre significatif très élevé, par exemple 1000 (msec). Dans ce cas, le paquet en essai de transmission et les paquets en file de sortie sur la ligne devront être redirigés vers une autre ligne de sortie. Au moment où la ligne redeviendra opérationnelle (des essais de



transmission se font pour cela à intervalles réguliers), les mêmes éléments de la table des délais seront décrémentés du même nombre.

Les modifications apportées dans la table des délais sont illustrées à la figure 3.2. Supposons que la table des délais d'un noeud soit dans la situation (a), (b) montre la modification lorsqu'un paquet est dirigé vers la ligne de sortie 3, (c) lorsque ce paquet a été émis et (d) lorsque la ligne a été déclarée non opérationnelle. (On suppose que la durée d'émission du paquet qui entre en OQ vaut 10 msec).

	1	2	3		1	2	3		1	2	3		1	2	3
1	15	12	30		15	12	40		15	12	30		15	12	1040
2	0	0	0		0	0	10		0	0	0		0	0	1010
3	30	20	10		30	20	20		30	20	10		30	20	1020
⋮															
⋮															
⋮															

(a)                      (b)                      (c)                      (d)

Fig. 3.2. exemple de modification de la table des délais.

Le délai d'attente estimé en file d'attente de sortie est donc très proche du délai réel puisqu'on compte réellement des durées d'émission des paquets en file d'attente.

Le cas des paquets spéciaux (paquets vides, paquets de routage) est particulier: puisqu'ils ne sont pas concernés par le routage, on ne s'intéresse pas à leur délai d'accès à la ligne de sortie puisqu'ils doivent de toute façon utiliser une ligne bien spécifique.

Néanmoins, on peut employer la même technique que pour les paquets normaux pour la modification de la table des délais, dès qu'ils ont été générés et placés sur la ligne de sortie.



La mise à jour de la table de routage du noeud peut se faire en fonction des modifications de la table des délais, comme suit: chaque fois qu'un paquet doit être dirigé vers une destination, on lui applique l'algorithme de sélection des routes, qui fournit, par l'intermédiaire de la table de routage recalculée, la ligne de sortie optimale vers la destination considérée.



### 3.5. La mise à jour externe.

Cette mise à jour signifie qu'un noeud informe ses voisins du contenu de sa table des délais par l'intermédiaire de paquets de routage.

Le voisin qui reçoit un paquet de routage modifie en conséquence sa table des délais.

Voyons tout d'abord à travers un exemple les deux moyens de calculer les vecteurs des délais minima; le second, inspiré de Cegrell (2), réduit la probabilité de formation de boucles. La figure 3.3. illustre un morceau de réseau et donne la partie relative de la table des délais du noeud 1.

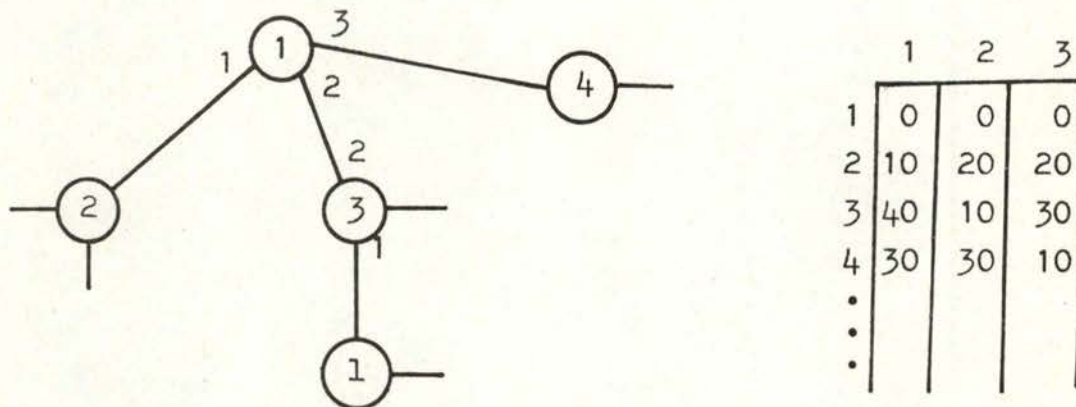


Fig. 3.3.

Quelle est l'information utile communiquée par le noeud 1 au noeud 3?

- 1) Si l'on suit la définition de  $VDM^1$  (cf. (2.3.)), le noeud 1 envoie au noeud 3:

$$(VDM^1(1), VDM^1(2), VDM^1(3), VDM^1(4), \dots)$$

où:

$$VDM^1(1) = 0$$

$$VDM^1(j) = TD^1(j, k_{opt}(j)) \quad j=2, \dots, N$$

donc

$$VDM^1 = (0, 10, 10, 10, \dots)$$

Ce vecteur est envoyé à tous les voisins de 1, en particulier 3.



- 2) On peut procéder autrement pour le calcul des VDM, en calculant un  $VDM^1$  propre à chaque voisin; on enverra ainsi au noeud 3 le vecteur  $VDM^1_3$  :  $VDM^1_3$  est le vecteur des délais minima estimés pour atteindre tous les noeuds à partir du noeud 1, sans utiliser la ligne de sortie vers le noeud 3, c'est-à-dire la ligne de sortie n° 2 du noeud 1 (concept d'horizon "interrompu"). Ce vecteur est défini comme suit:

$$VDM^1_3(1) = 0$$

$$VDM^1_3(j) = \min_{\substack{k=1, \dots, NLS(1)=3 \\ k \neq 2}} TD^1(j, k), \quad j=2, \dots, N$$

on obtient ainsi

$$VDM^1_3 = (0, 10, 30, 10, \dots)$$

Cette façon de calculer les VDM permet de réduire la possibilité de formation de boucles: lorsque la meilleure route pour atteindre 1 à partir de 1 passe par 3, on a:

$$VDM^1_3(1) > VDM^1(1)$$

ce qui réduit la probabilité que la meilleure route pour atteindre 1 à partir de 3 passe par 1 (auquel cas une boucle  $3 \Rightarrow 1$  se formerait).

La figure 3.4. donne un exemple de mise à jour au noeud 3 lorsqu'il reçoit  $VDM^1_3$ . La première colonne montre le  $VDM^1_3$  reçu) la seconde les valeurs actuelles de la grandeur interne  $W^3$  pour la ligne de sortie 2 du noeud 3, la troisième la somme ligne à ligne des 2 premières; puis vient la table des délais du noeud 3 avec la colonne mise à jour (colonne correspondant à la ligne vers le noeud 1, soit la ligne n° 2).



$VDM_3^1$	$W^3(2)$	$\Sigma$	1	2	
0	17	17		17	1
10	17	27		27	2
30	17	47		17	3
10	17	27		27	4
	.	.			
	.	.			
	.	.			

Fig. 3.4.

L'entrée pour le noeud 3 n'est pas nulle mais vaut 17, c'est-à-dire le délai d'attente dans la file de la ligne 2; dans la mise à jour interne au noeud 3, on ne tient pas compte de  $VDM_3^1(3)$ , délai estimé d'un transfert de 1 vers 3 sans utiliser la ligne entre les 2 noeuds qui n'est évidemment d'aucune utilité au noeud 3.

Lorsqu'un noeud reçoit un VDM, il effectue immédiatement la mise à jour de la colonne correspondante de la table des délais.

Pour la généralisation de la définition, soit  $VDM_j^k$  le vecteur des délais minima formé au noeud  $k$  et destiné au noeud  $j$ , via la ligne de sortie  $m$  du noeud  $k$ .

Ce vecteur est défini comme suit:

$$VDM_j^k(i) = \min_{\substack{l=1, \dots, NLS(k) \\ l \neq m}} TD^k(i, l) \quad \begin{matrix} i=1, \dots, N \\ i \neq k \end{matrix} \quad (3.1)$$

$$VDM_j^k(k) = 0$$

Notons  $VDM_{rj}^k$  le vecteur des délais minima reçu au noeud  $k$  sur la ligne  $j$  et  $W^k(j)$  l'estimation du délai d'accès à la ligne de sortie  $j$  du noeud  $k$ .

La mise à jour de la colonne  $j$  de la table des délais du noeud  $k$  se fait comme suit:



$$\begin{aligned}
 TD^k(i, j) &= VDM_{rj}^k(i) + W^k(j) & i=1, \dots, N; i \neq k \\
 TD^k(k, j) &= W^k(j)
 \end{aligned}
 \tag{3.2}$$

Cette deuxième façon de calculer les VDM est sans influence sur l'utilisation de la ligne pour la transmission des paquets de routage, par rapport à la première façon. Mais elle demande le calcul de  $NLS(i)$  vecteurs pour le noeud  $i$  au lieu de 1 vecteur dans l'autre cas. En général  $NLS(i)$  vaut au maximum 4 ou 5, pour tout  $i$ .

Un autre problème important est de déterminer l'instant auquel la mise à jour externe doit intervenir, ou plutôt celui auquel le VDM doit être calculé et transmis, la mise à jour de la table des délais s'effectuant lorsqu'un noeud reçoit un VDM d'un noeud voisin.

En toute logique, l'information de routage (paquets contenant les VDM) doit d'une part, utiliser les lignes au minimum (pour que la capacité utile réelle des lignes soit la plus grande possible) et d'autre part, être suffisamment fréquente pour toujours représenter une situation valable de l'état actuel du sous-réseau, ce qui est indispensable pour assurer que la route que l'on va choisir est encore la meilleure et pour éviter une formation de boucle due à un retard ou à une absence de transmission d'information d'échange.

Ces deux critères sont à première vue contradictoires: une information plus fréquente utilisera plus la ligne. Il faut donc employer une technique qui satisfasse au mieux ces deux critères.

Plusieurs possibilités d'action se présentent, elles sont regroupées dans le tableau de la figure 3.5. Elles sont classées suivant le type de calcul du VDM (instantané, par moyenne, arithmétique ou "pondérée") et son mode de transmission (immédiat, périodique ou aperiodique).

Les numéros repris dans le tableau renvoient aux paragraphes des pages suivantes.



type de calcul du VDM mode de trasmission	instantané	par moyenne arithmétique	par moyenne "pondérée"
immédiat	3.5.1.1		
périodique	3.5.1.2	3.5.2.1	3.5.3.1
apériodique	3.5.1.3		3.5.3.2

Fig. 3.5

### 3.5.1 Calcul du VDM par une mesure instantanée

Le noeud calcule son VDM ( ou ses VDM si l'on choisit la possibilité 2) du paragraphe 3.5 ) par une mesure instantanée de sa table des délais.

#### 3.5.1.1 Mode de trasmission immédiat

Ce mode de transmission concerne le cas particulier où une ligne devient non opérationnelle: le noeud qui découvre ce fait ( les éléments d'une colonne de sa table des délais atteignent une valeur supérieure à 1000 ) devra aussitôt former le ou les VDM concernés et les transmettre à ses voisins: l'événement a déclenché le calcul immédiat et la transmission immédiate du (des) VDM.

En effet, il est primordial que cette information sur une panne de ligne soit propagée au plus tôt dans tout le sous-réseau. Un noeud qui reçoit un VDM comportant des éléments supérieurs à 1000 met à jour sa table des délais et devra lui aussi transmettre immédiatement le (les) VDM affectés par cette panne de ligne.

Le mode immédiat augmente la capacité d'adaptation de l'algorithme de routage face à des pannes de ligne. Il devrait figurer dans tout algorithme de routage adaptatif distribué.



### 3.5.1.2. Mode de transmission périodique.

Le noeud calcule ses VDM à instants fixes (toutes les  $T_u$  secondes) et les transmet alors à ses voisins: il extrait les VDM de sa table des délais, met en forme les paquets de routage contenant ces VDM et les place dans les files de sortie des lignes correspondantes.

On a donc le schéma suivant (fig. 3.5.)

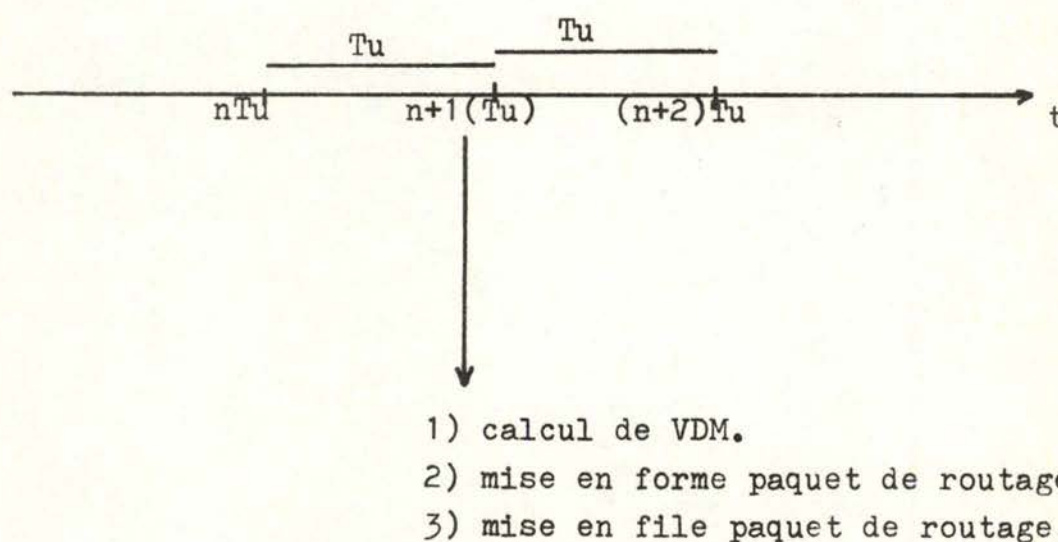


Fig. 3.6. Mode de transmission périodique.

Cette façon de procéder est simple et rapide à mettre en oeuvre, elle ne tient pas compte des variations de la table des délais entre deux instants  $n Tu$  et  $(n+1) Tu$ . L'efficacité dépend donc seulement de la longueur de la période  $T_u$  qui sépare deux mesures: elle sera d'autant meilleure que  $T_u$  sera petit, surtout pendant une période de variations rapides, mais on envoie alors les paquets de routage avec une plus grande fréquence, ce qui diminue la capacité utile des lignes. Cette méthode a été employée dans Arpa.



### 3.5.1.3. Mode de transmission aperiodique.

Le noeud fait un échantillonnage de sa table des délais toutes les  $T_e$  secondes.  $T_e$  étant déterminé par la fréquence  $f_e = 1/T_e$  d'échantillonnage minimum que l'on détermine (en fonction de diverses conditions). Si un des éléments de la table des délais qui intervient dans un VDM subit une variation supérieure en module à un certain seuil  $S$ , par rapport à la valeur de sa mesure précédente (conservée en mémoire du noeud), on forme le VDM ou les VDM qui sont concernés par cette variation et on les transmet aux noeuds correspondants. La figure 3.7. illustre cette situation.

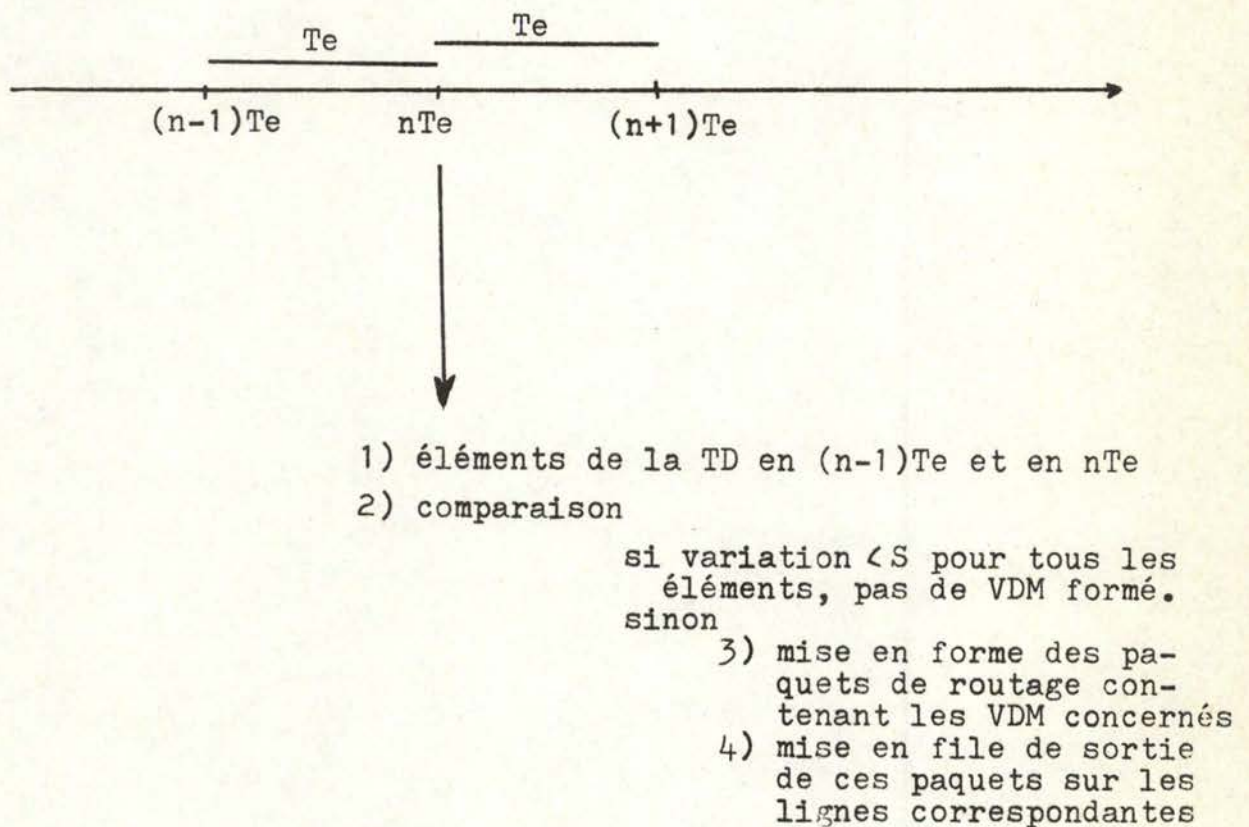


Fig. 3.7.

Ici, on envoie de l'information aux noeuds voisins lorsque cela est nécessaire; la ligne n'est utilisée que pour de l'information utile. Si les conditions de charge sont peu impor-



tantes, la table des délais sera modifiée plus rarement et on enverra moins souvent de l'information aux voisins.

Nous nous intéresserons au chapitre 4 au problème de la détermination d'une fréquence d'échantillonnage. La façon dont l'on détermine  $T_e$  permet d'affirmer que sur une période  $T_e$ , les variations seront en général peu importantes puisque la fréquence d'échantillonnage  $f_e$  est déterminée pour suivre au mieux l'évolution de la table des délais ( $T_e \ll T_u$ ). Comme nous le verrons dans l'analyse de la simulation, cette fréquence est déterminée en fonction de l'évolution de la grandeur interne  $W$  c'est-à-dire en fonction de l'évolution des files d'attente de sortie. On tiendra donc ainsi compte de l'évolution de la table des délais due aux variations de  $W$ , ce qui n'était pas le cas dans le point 3.5.1.2. pendant toute la période  $T_u$ . D'autre part, on tiendra aussi compte des VDM venant des noeuds voisins et pouvant arriver dans n'importe quelle période  $T_e$ .

Une amélioration peut encore être apportée à la méthode expliquée ci-dessus. En effet, lorsqu'on compare la table des délais de chaque paire d'instantanés consécutifs  $nT_e$  et  $(n+1)T_e$ ; la variation peut être inférieure à  $S$  pendant une très longue période. Si les variations se font presque toujours dans le même sens, leur accumulation peut amener une variation totale depuis le dernier instant de formation de VDM supérieure à  $S$ .

On compare alors la table des délais actuelle avec celle que l'on avait lorsqu'on a formé des VDM pour la dernière fois (chaque fois que l'on a formé un VDM, on conserve la table des délais en mémoire jusqu'au moment où l'on reformera des VDM).

Il se peut aussi que sur une période  $T_e$ , la table des délais n'ait pas été modifiée; à ce moment, il est inutile de faire la comparaison. Il suffit pour réaliser cela de positionner quelle que part un bit  $B$  à 1 lorsque la table des délais est modifiée; lorsqu'on arrive à l'instant d'échantillonnage suivant, on effectue la comparaison et les opérations nécessaires suivant le résultat, puis on positionne le bit  $B$  à 0; si le bit est positionné à 0 lorsqu'on arrive à un instant d'échantillonnage,



on n'effectue pas l'échantillonnage ni la comparaison puisqu'aucune modification de la table des délais n'est apparue depuis l'instant d'échantillonnage précédent.

Cette situation est illustrée à la fig. 3.8.

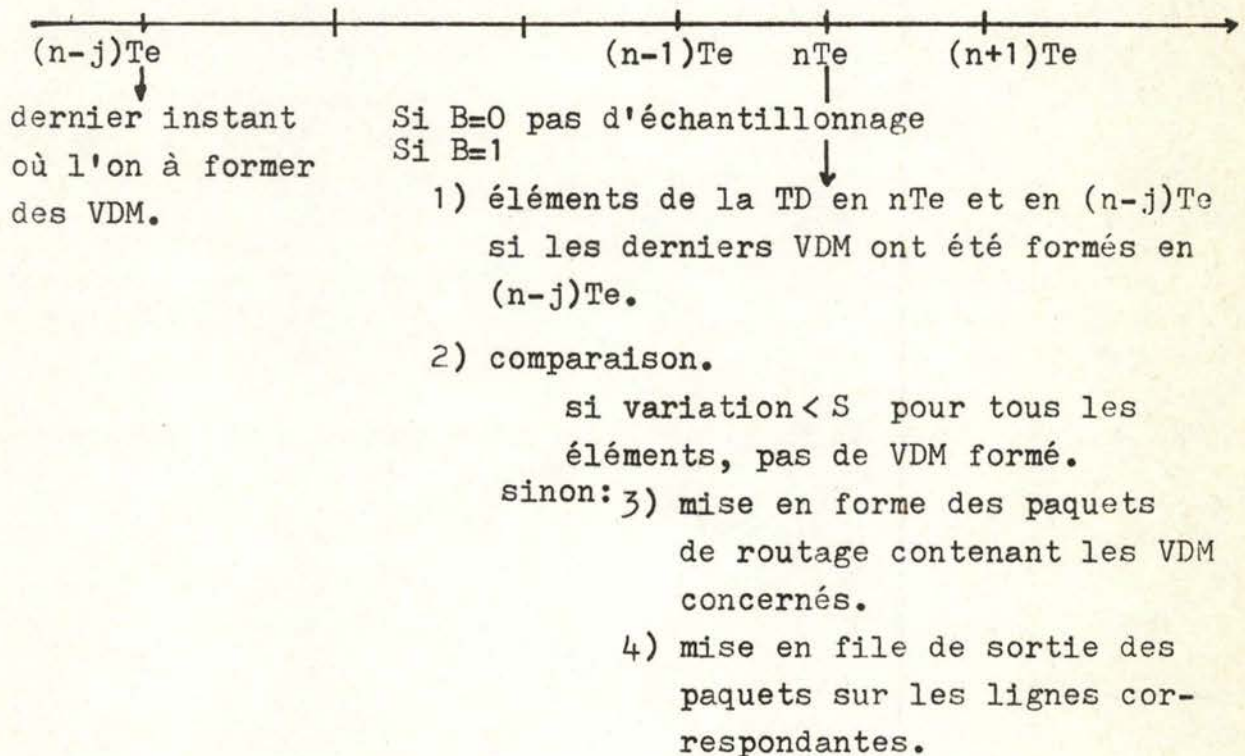


Fig. 3.8.

Par rapport au mode périodique, le mode apériodique demande des comparaisons à chaque étape et suit mieux les variations dues à  $W$  (on peut s'attendre à ce que la période  $Te$  soit assez bien plus courte que la période  $Tu$ ).







A chaque période  $T_e$  on prélève un échantillon formé suivant la formule employée pour calculer le VDM (ou les VDM) et on calcule la moyenne arithmétique actuelle.

Si l'on désigne par  $f(T_i^n)^{(1)}$  la valeur de la mesure du  $i^{\text{ème}}$  échantillon de la  $n^{\text{ème}}$  période  $[(n-1)Tu, nTu]$  et par  $\bar{M}^n$  la moyenne arithmétique des  $m$  échantillons de cette  $n^{\text{ème}}$  période, on a:

$$\bar{M}^n = \frac{1}{m} (f(T_1^n) + f(T_2^n) + \dots + f(T_m^n))$$

$$\bar{M}^n = \frac{1}{m} (f((n-1)Tu + h) + f((n-1)Tu + h + Te) + f((n-1)Tu + h + 2Te) + \dots + f((n-1)Tu + h + mTe)) \quad h < Te$$

A tout moment, si l'on désigne par  $\bar{M}_k^n$  la moyenne des  $k$  premiers échantillons ( $1 \leq k \leq m$ ), elle s'exprime en fonction de la moyenne des  $(k-1)$  premiers échantillons par:

$$\bar{M}_k^n = \frac{(k-1) \bar{M}_{k-1}^n + f(T_k^n)}{k}$$

$$\bar{M}_1^n = f(T_1^n)$$

et on a:

$$\bar{M}^n = \bar{M}_m^n \quad (\text{pour tout temps de } ((n-1)Tu, nTu)).$$

Il suffit donc de conserver en mémoire la moyenne actuelle.

Une nouvelle moyenne est transmise toutes les  $T_u$  secondes.

-----

(1)  $f(T_i^n)$  représente l'équivalent du VDM (ou des VDM) dont la formule est donnée en 2.3. (ou en 3.1.)



### 3.5.3. Calcul du VDM par moyenne "pondérée".

L'information échangée entre noeuds voisins reflète une situation moyenne de la situation du réseau sur une période, mais la moyenne transmise n'est plus une moyenne arithmétique. Cette moyenne est recalculée à une fréquence  $f_e = 1/T_e$  mais est transmise soit à la fin de chaque période  $T_u$ , soit lorsque elle subit une variation supérieure à un certain seuil.

#### 3.5.3.1. Mode de transmission périodique.

Les VDM formés des moyennes pondérées des  $m$  échantillons prélevés sur une période  $T_u$  sont transmis à la fin de la période  $T_u$ .

La fig. 3.10 illustre cette situation.

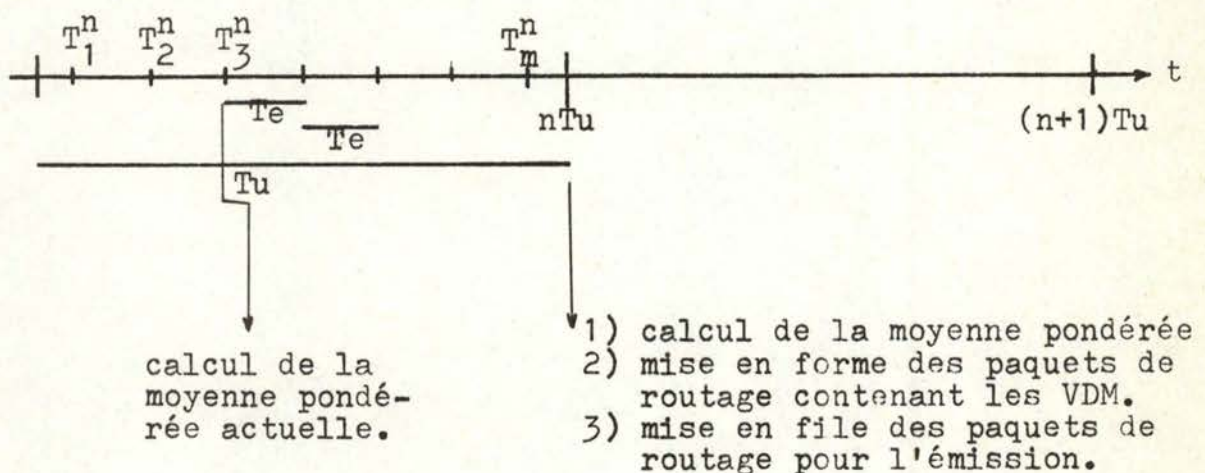


Fig. 3.10

Soit  $N_j^n$  la moyenne pondérée calculée à l'instant  $T_j^n$  de la  $n^{\text{ème}}$  période de mesure et  $N^n$  la moyenne pondérée en fin de période  $((n-1)T_u, nT_u)$

On définit:



$$\bar{N}^n|_j = a \bar{N}^n|_{j-1} + (1-a) f(T_j^n) \quad j \neq 1 \quad a \in ]0,1[$$

$$\bar{N}^n|_j = a \bar{N}^{n-1}|_{m_{n-1}} + (1-a) f(T_1^n). \quad j=1$$

Si  $m_{n-1}$  est le nombre d'échantillonnages dans la  $(n-1)$ <sup>ième</sup> période de longueur  $T_u$ .

$$\bar{N}^n = \bar{N}^n|_{m_n}$$

Cette moyenne est transmise toutes les  $T_u$  et est calculée au début d'une période en fonction de sa valeur à la fin de période précédente.

La façon dont on tient compte du passé dépend du paramètre  $a$  : plus il sera petit, plus on donnera un poids important à la dernière mesure effectuée. Tout au long de la période cette moyenne évolue suivant les modifications de la table des délais, alors que la moyenne arithmétique est évidemment constante sur une période.

### 3.5.3.2. Mode de transmission aperiodique.

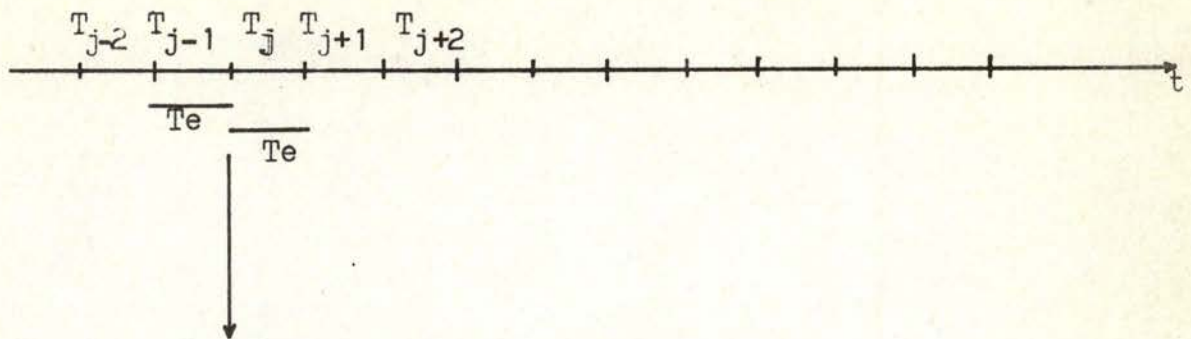
La moyenne évolutive est calculée de la même façon que dans le cas précédent, soit  $\bar{N}|_j$  la moyenne pondérée calculée à l'instant  $T_j$ .

$$\bar{N}|_j = a \bar{N}|_{j-1} + (1-a) f(T_j) \quad a \in ]0,1[$$

On transmet les VDM correspondants à une moyenne pondérée lorsque la moyenne pondérée calculée en un instant d'échantillonnage  $T_j$  subit une variation supérieure à un seuil  $S_2$  par rapport à sa valeur précédente. Pour déterminer ce seuil, il faut tenir compte de la valeur de  $a$  (si  $a$  est grand, les variations seront relativement petites).

La figure 3.11. illustre cette situation.





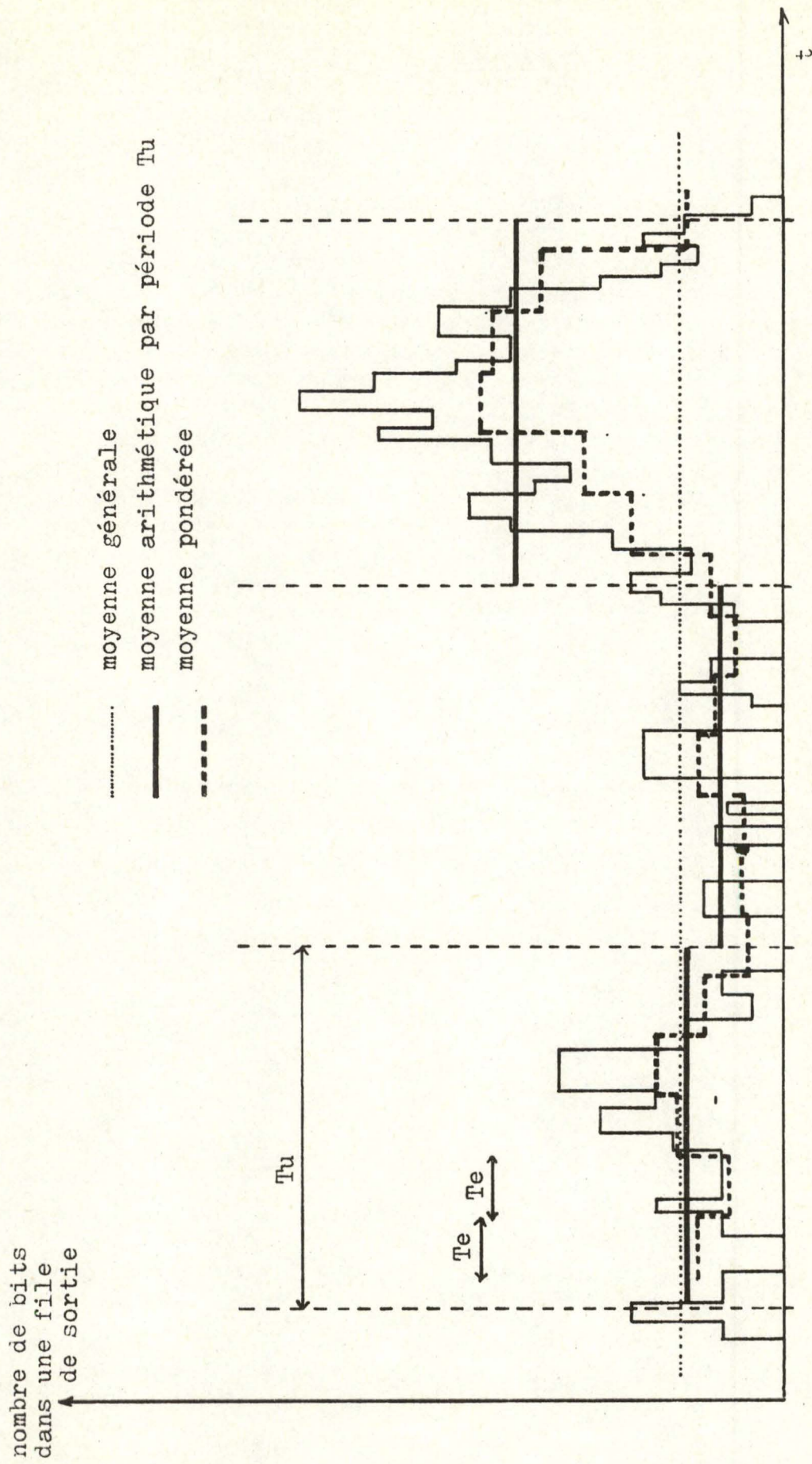
- 1) calcul de la moyenne pondérée.
- 2) comparaison avec la valeur précédente.
  - si variation < S2 : pas de VDM.
  - si variation > S2 :
    - 3) mise en forme des paquets de routage contenant les VDM.
    - 4) mise en file des paquets de routage pour l'émission.

Fig. 3.11.

L'avantage par rapport à une moyenne pondérée transmise périodiquement est que l'on transmet des VDM lorsqu'une variation importante intervient, alors que dans l'autre cas elle peut varier assez bien pendant une période pour revenir en fin de période à une valeur proche de sa valeur à la fin de la période précédente.

La figure 3.12 compare les différentes moyennes décrites en fonction de la modification des éléments d'une colonne de la table des délais, due à la variation de la longueur de la file de sortie correspondante, longueur obtenue par la somme des bits des paquets se trouvant dans la file d'attente.





Fig;3.12 Comparaison des types de moyenne



### 3.6. L'accessibilité.

L'algorithme d'accessibilité présenté au paragraphe 2.3.1.f utilise les vecteurs MHV qu'il faut transmettre, sous forme de paquets de routage, aux noeuds voisins.

Il est possible de faire l'économie de ces paquets de routage contenant les MHV, étant entendu que l'échange des VDM est de toute façon indispensable pour la sélection des routes.

Lorsqu'un noeud  $j$  devient inaccessible pour le noeud  $p$ , les délais  $TD^P(j,k), \forall k=1, \dots, NLS(p)$  se mettent à diverger, tout comme la distance minimale de  $P$  à  $j$  évaluée en "hops".

Alors lorsque les délais  $TD^P(j,k), \forall k$ , dépasseront un seuil  $SC$ , le noeud  $j$  pourra être considéré comme inaccessible à partir de  $P$ .

Lors de la sélection d'une ligne de sortie vers la destination  $j$ , si l'élément minimal de la  $j^{i\text{ème}}$  ligne de la table des délais  $TD^P$  est supérieure à  $SC$ , le paquet sera détruit car le noeud de destination est inaccessible.

Ici encore, l'algorithme d'accessibilité est confondu avec l'algorithme de sélection de routes: lorsqu'on choisit une route (formation de la table de routage), on a tenu compte directement des noeuds non accessibles puisque ce renseignement est contenu dans la table des délais.



### 3.7. Conclusions.

La mise à jour interne peut refléter assez fidèlement la charge sur une ligne, à travers l'évolution de la file d'attente de sortie de cette ligne, si l'on tient compte de la longueur des paquets s'y trouvant. Le comportement de l'algorithme face à des pannes de ligne ou de noeud est simple: au moment de la sélection d'une route, les renseignements sur l'accessibilité à un noeud et sur la possibilité d'utilisation d'une ligne se trouvent dans la table des délais. Ces renseignements seront propagés au plus vite dans le sous-réseau, ce qui est indispensable, si l'on utilise le mode immédiat pour la transmission des VDM. Nous avons vu qu'une deuxième manière de calculer les VDM d'un noeud permet de réduire fortement la possibilité de formation de boucles. Elle demande seulement, par rapport à la première, le calcul de plusieurs vecteurs, un pour chaque ligne de sortie. Mais en général, le nombre de lignes de sortie d'un noeud est très limité.

Que l'on transmette une mesure instantanée ou moyenne aux noeuds voisins, il est toujours préférable de tenir compte d'une variation d'une certaine importance et de la transmettre aux noeuds voisins; l'adaptation à la charge actuelle n'en sera que meilleure.

Dans ce cas, le problème est de déterminer une fréquence d'échantillonnage, en fonction de l'évolution des files de sortie et à partir de celle-ci, une période d'échantillonnage moyenne.

Le mode de transmission périodique, bien que plus facile à implémenter, ne semble pas satisfaisant; la période  $T_u$  de longueur invariable peut être beaucoup trop longue dans les cas de forte charge.

Nous avons d'ailleurs vu que dans le réseau Arpa où l'on utilise le mode périodique, on a du adjoindre des techniques à l'algorithme de routage pour limiter la formation de boucles, mais au dépend de l'adaptation à la charge du réseau.



Le chapitre suivant nous donne des renseignements sur l'évolution dynamique d'un système d'échange entre deux noeuds.



ANALYSE ET RESULTATS DE LA SIMULATION4.1. Utilisation de la simulation.

Bien qu'un modèle analytique d'un système à étudier soit souvent préféré à une simulation, il n'est pas toujours possible de construire un tel modèle pour représenter valablement le système. Les modèles stochastiques liés à la théorie des files d'attente peuvent donner des résultats dans des cas généraux. Mais lorsque le système à modéliser est très particulier et comprend de nombreux paramètres, on ne peut pas utiliser un modèle analytique sans imposer des hypothèses trop restrictives qui font que le modèle reste très éloigné du système réel.

La simulation permet d'étudier le comportement macroscopique du système à étudier, d'observer les occurrences réelles des événements. Elle donne donc un aperçu du comportement physique du système et permet d'entreprendre des expériences sur le système pour obtenir de l'information sur les paramètres critiques du problème.

On peut considérer que la simulation est une manière d'utiliser un ordinateur pour produire un modèle raisonnable du système à étudier. Ce modèle est un programme; en exécutant celui-ci, l'ordinateur émule le comportement du système et accumule des informations au fur et à mesure de la progression de l'exécution du programme; il produit une sortie à partir de laquelle le comportement du modèle peut être déduit.

Chaque fois que cela sera possible, on comparera les résultats obtenus par simulation aux résultats donnés par la solution analytique.

Les programmes sont écrits en FORTRAN IV, ce qui demande d'y introduire le concept d'horloge, de génération de nombres aléatoires. La simulation est "orientée-événements", les événements significatifs du système guidant la progression de l'horloge.



Après divers essais de programmes de simulation de files d'attente, de génération de nombres aléatoires, nous avons utilisé deux programmes de simulation d'un système formé de deux noeuds et d'un générateur envoyant des paquets à ces 2 noeuds. Le premier ne tient compte que du trafic dans une direction de la ligne full-duplex entre les 2 noeuds (il est étudié au point 4.2.). Le second tient compte du trafic inverse (il est étudié au point 4.3. et décrit en annexe 2). Dans ces programmes, la simulation comme nous l'avons dit est "orientée événements", les événements étant les instants significatifs repris à la figure 4.2.

L'idée de base étant, après un traitement quelconque, de traiter le paquet dont un des temps significatifs est actuellement le plus petit. Le traitement résultant nous amenant à l'instant significatif suivant de ce paquet.

Ce type d'exécution demande en Fortran un jeu de pointeurs assez complexe. Dans certains cas, la priorité étant donnée à la transmission des blocs ACK par des paquets vides, il faut après le choix du temps significatif minimum, effectuer un retour en arrière pour s'assurer qu'il n'y a pas (ou qu'il y a) un traitement prioritaire à effectuer. Ces cas distincts étant assez nombreux, le programme devient relativement complexe.

La mise au point des programmes a demandé un temps assez important en raison de cette complexité.

L'analyse des résultats se fait à partir de diverses routines de calcul de moyenne, écart-type, distribution cumulative et à partir de compteurs introduits dans le programme et suivant l'évolution de l'exécution.

La collecte des résultats nécessaires à la réalisation des graphiques qui donnent le temps passé en file d'attente (par l'intermédiaire du nombre de bits en file d'attente) et le nombre de paquets en file d'attente s'est faite à partir de l'impression en vrac des temps significatifs de chacun des paquets.

Les paramètres sont lus en début de programme.



A partir de ceux-ci, des sous-programmes calculent et mettent en mémoire les heures d'arrivées et les longueurs de chaque paquet, en utilisant le générateur. Après cela, la simulation proprement dite est exécutée.



## 4.2. Simulation entre 2 noeuds sans trafic inverse.

### 4.2.1. Le programme de simulation.

Le programme a pour but d'étudier le comportement des files d'attente d'un noeud à travers la liaison entre 2 noeuds, liaison qui consiste en une ligne full-duplex et en un ensemble de files d'attente.

Le programme comporte 3 parties:

- un générateur de paquets simulant l'arrivée des paquets dans la file d'entrée IQ du noeud 1 (file d'attente devant le serveur "traitement du paquet").
- un programme de simulation, suivant le protocole C pour la routine noeud-noeud (c'est à dire que les ACK sont contenu dans des paquets de contrôle ou paquets "vides").
- un ensemble de sorties permettant l'analyse statistique des résultats.

Le système simulé est celui de la figure 4.1.

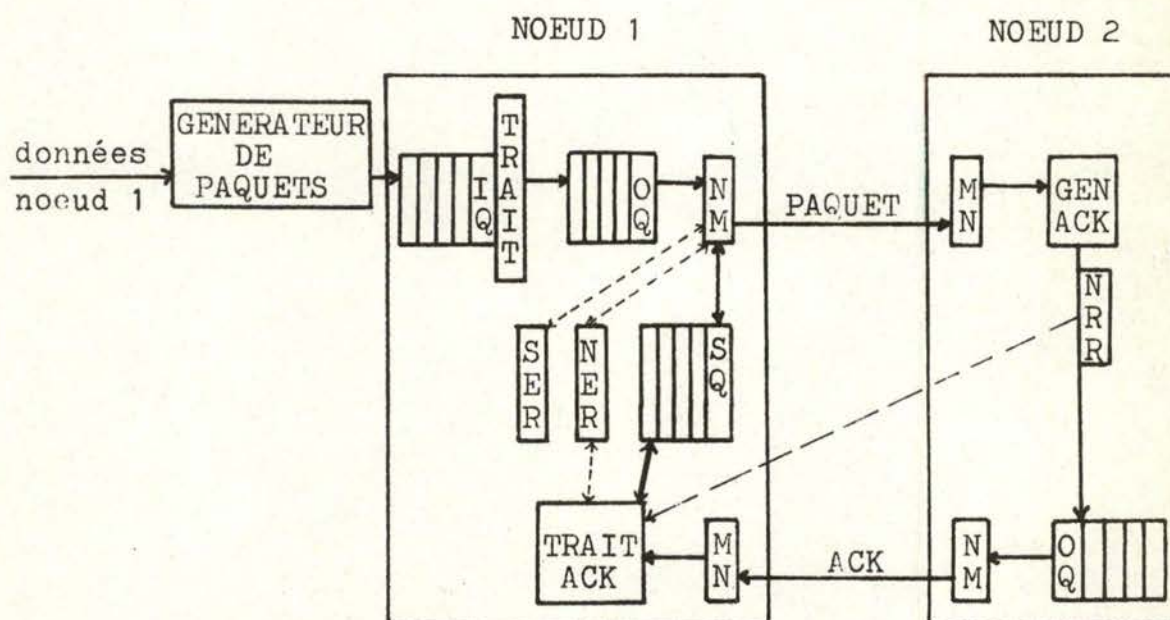


Fig. 4.1.

Le fait de ne pas envisager de trafic inverse signifie que le paquet contenant un ACK sera émis dès la génération de l'ACK au noeud 2.

### Le générateur de paquets.

Pour l'ensemble des lignes autre que la ligne étudiée, le générateur de paquets produit une table contenant l'heure d'arrivée  $t_1$  du paquet (heure d'interruption de la routine modem-noeud) et une table contenant la longueur totale du paquet.

On supposera que tous les paquets sont destinés à la file OQ étudiée.

Les arrivées suivent un processus de Poisson de taux  $k$  (paquets/seconde); le temps inter-arrivée a donc une distribution exponentielle de moyenne  $1/k$  (seconde).

La longueur totale LP d'un paquet est donné par:

$$LP = h + s + LT$$

$$LT = a + b$$

où  $h$  est la longueur de l'overhead hardware (caractères nécessaires à la transmission.)

$s$  est l'en-tête software.

$LT$  est la longueur du texte.

$a$  est une constante.

$b$  est une variable de distribution exponentielle tronquée pour assurer  $LT \leq LT_{\max}$ .

### Les temps significatifs.

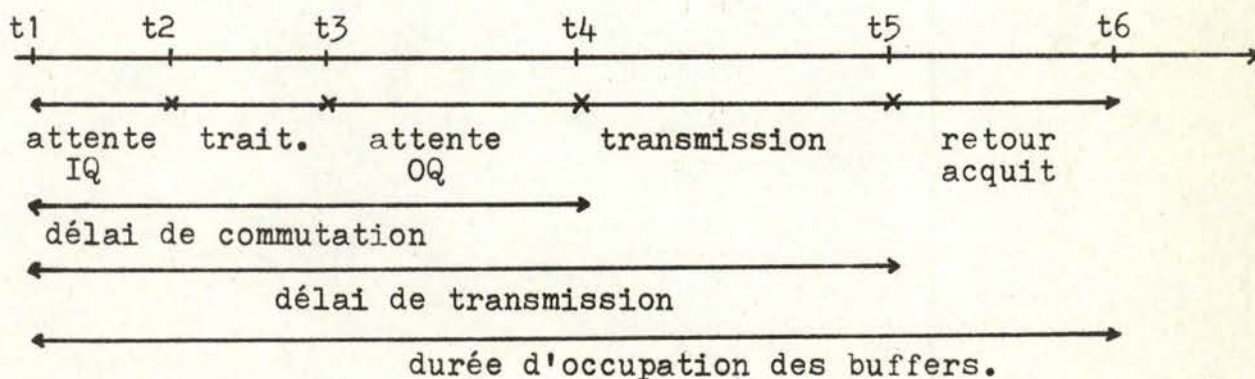


Fig. 4.2.



Les conditions de test.

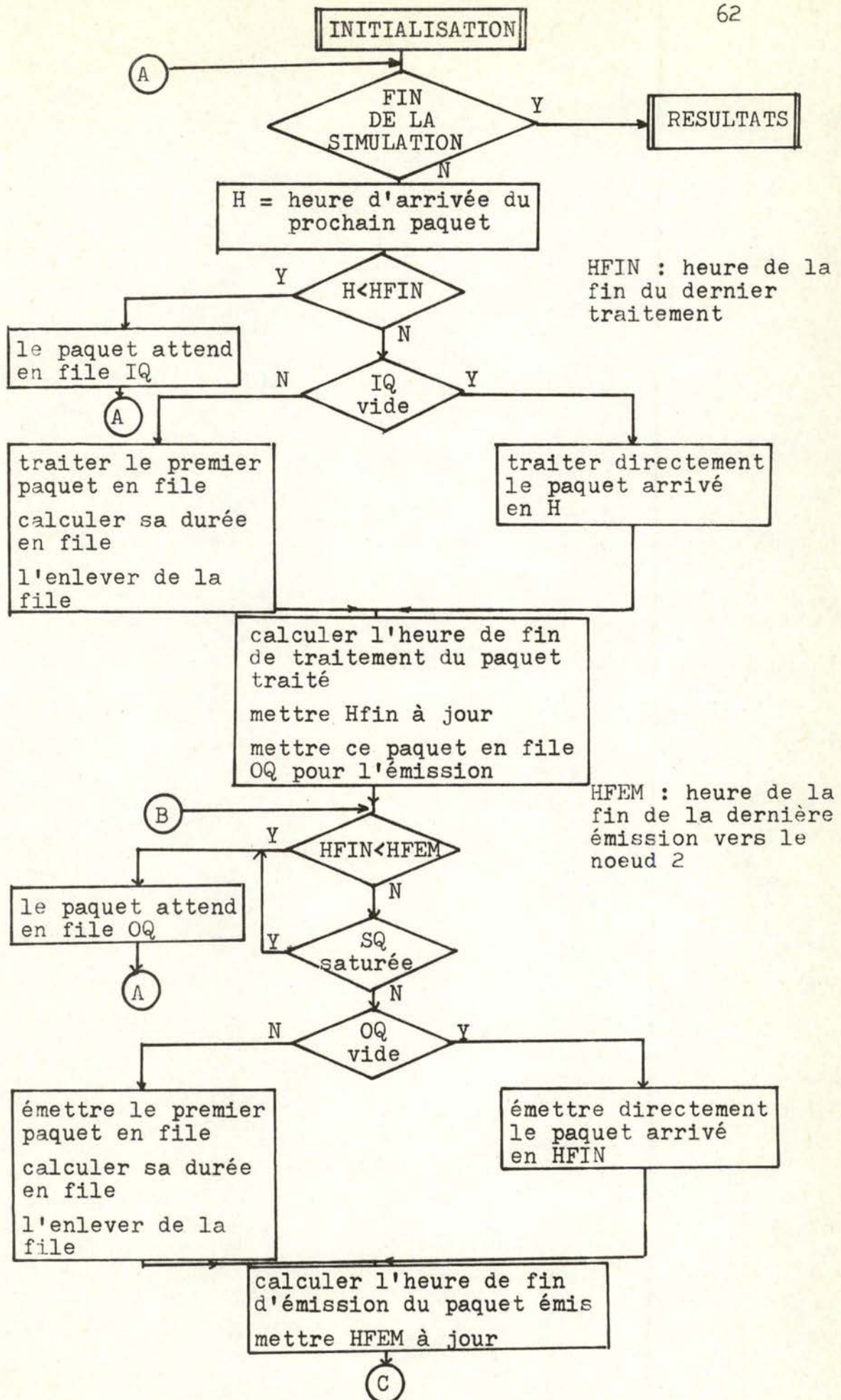
Les divers essais effectués ont été réalisés sous les conditions suivantes:

- 1) La capacité en buffer n'est pas limitée, sauf bien sûr pour la SQ qui est limitée à 8
- 2) Chaque paquet est reçu correctement .
- 3) Pour chaque ligne arrivant au noeud 1, on a adopté la même longueur moyenne  $\overline{LT} = 512$  bits (moyenne de la distribution non tronquée);  $a = 0$  et  $LT_{\max} = 1024$  bits. Le taux d'arrivée au noeud 1 varie d'un essai à l'autre.
- 4) La capacité de la ligne est 48 kbits/s (ou 19.2 kbits/s); le temps d'exécution de la routine modem-noeud et de la routine noeud-modem est de 0.5 msec et le temps d'exécution de la routine "tâche principale" (traitement d'un paquet) est de 5 msec.
- 5) Le délai de propagation sur la ligne est négligeable, l'entête software  $s = 80$  bits et l'overhead hardware  $h = 72$  bits
- 6) La simulation porte sur 1000 paquets.

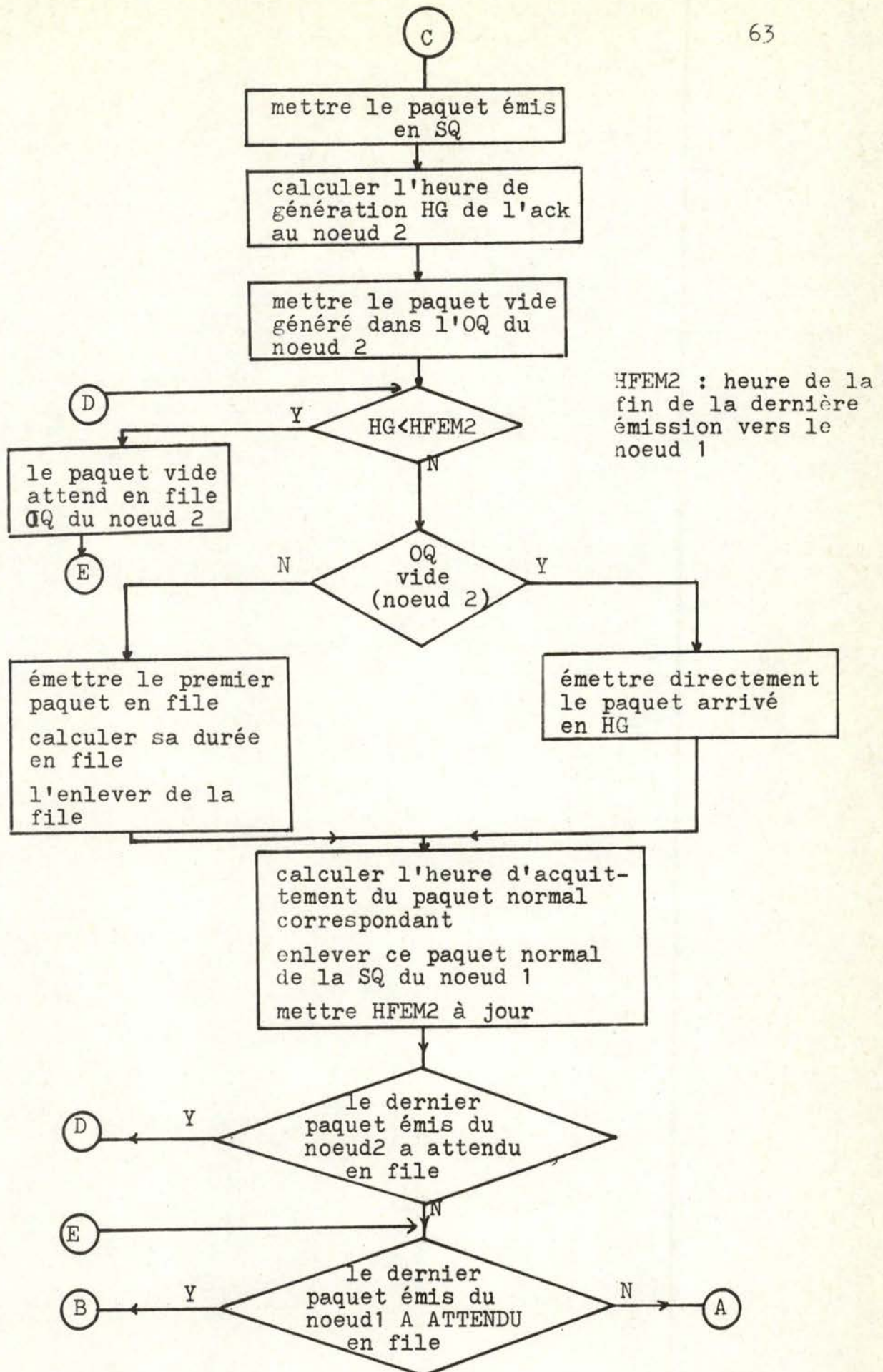
A partir d'une exécution de la simulation, on peut obtenir les différentes heures  $t_2$  à  $t_6$  et les délais montrés à la figure 4.2, pour chaque paquet; ainsi que les moyennes, écart-type, distributions cumulatives, valeurs maximales.

On obtient aussi des résultats sur les longueurs des diverses files IQ, OQ, SQ, en nombre de paquets et en nombre de bits.

Un organigramme retrace l'évolution du programme de simulation dans ses grandes lignes aux deux pages qui suivent.







ESSAIS	1	2	3	4	5
capacité ligne (kb/s)	48.0	48.0	48.0	48.0	19.2
taux d'arrivée (paq/s)	30.0	50.0	70.0	100.0	30.0
durée d'attente en I <sub>Q</sub> (ms)					
moyenne	0.44	0.83	1.37	2.52	0.44
écart-type	1.27	1.87	2.56	3.85	1.27
maximum	10.7	12.5	15.8	24.2	10.7
durée d'attente en O <sub>Q</sub> (ms)					
moyenne	2.30	5.29	12.70	141.79	42.58
écart-type	5.28	8.56	15.24	83.08	47.67
maximum	44.3	54.2	94.2	343.3	288.4
durée moyenne de transmission (ms)	10.40	10.40	10.40	10.40	26.01
durée moyenne de retour de l'acquit (ms)	2.67	2.67	2.67	2.67	5.17
délai moyen de commutation (ms)	7.74	11.12	19.08	149.37	48.02
délai moyen de transmission (ms)	18.13	21.52	29.48	159.77	74.03
durée moyenne d'occupation des buffers (ms)	20.81	24.19	32.15	162.44	79.19
durée de la simulation (ms)	34455	20674	14768	10339	34455

Tableau 4.1. Résultats au noeud 1: délais.



ESSAIS	1	2	3	4	5
capacité ligne (b/s)	48.0	48.0	48.0	48.0	19.2
taux d'arrivée (paq/s)	30.0	50.0	70.0	100.0	30.0
nombre de paq. en IQ					
moyenne	0.01	0.04	0.09	0.25	0.01
maximum	3	3	3	5	3
nombre de bits en OQ					
moyenne	110	254	610	6805	817
maximum	3105	3556	5256	17406	6225
nombre de paq. en OQ					
moyenne	0.07	0.26	0.85	13.54	1.23
maximum	4	6	8	31	9
nombre maximum de paq. en SQ	1	1	1	1	1
nombre maximum de paq. en (OQ+SQ)	5	7	9	32	10

Tableau 4.2. Résultats au noeud 1: files d'attente.

Les tableaux 4.1. et 4.2. reprennent les principaux résultats statistiques obtenus dans les divers essais.

a) le générateur de paquets.

Nous avons supposé que le temps interarrivée suit une distribution exponentielle (hypothèse de Poisson). A la fig. 4.4., la distribution cumulative de ce temps interarrivée résultant du générateur de paquets est comparé avec la distribution cumulative théorique obtenue par l'hypothèse de Poisson. Le comportement du générateur est donc très satisfaisant, les petites valeurs sont très légèrement inférieures et les grandes valeurs supérieures.

Le fait de tronquer la distribution des longueurs de texte des paquets donne une moyenne inférieure à la moyenne de la distribution non tronquée. Si  $1/k$  est la moyenne de la distribution non tronquée, la moyenne de la distribution tronquée en  $L_{\max}$  vaut:

$$\begin{aligned} \overline{LT}_{\text{tronq}} &= \int_0^{L_{\max}} kt e^{-kt} dt \\ &= k \left( \frac{t e^{-kt}}{-k} \right) \Big|_0^{L_{\max}} + \frac{1}{k} \int_0^{L_{\max}} e^{-kt} dt \\ &= -e^{-kt} \left( t + \frac{1}{k} \right) \Big|_0^{L_{\max}} \end{aligned}$$

Dans notre cas,  $L_{\max} = 1024$  bits.

Ainsi, si la moyenne  $\frac{1}{k} = 256$  bits, on obtient:

$$\overline{LT}_{\text{tronq}} = 256 (1 - 5 e^{-4}) = 232.56 \text{ bits}$$



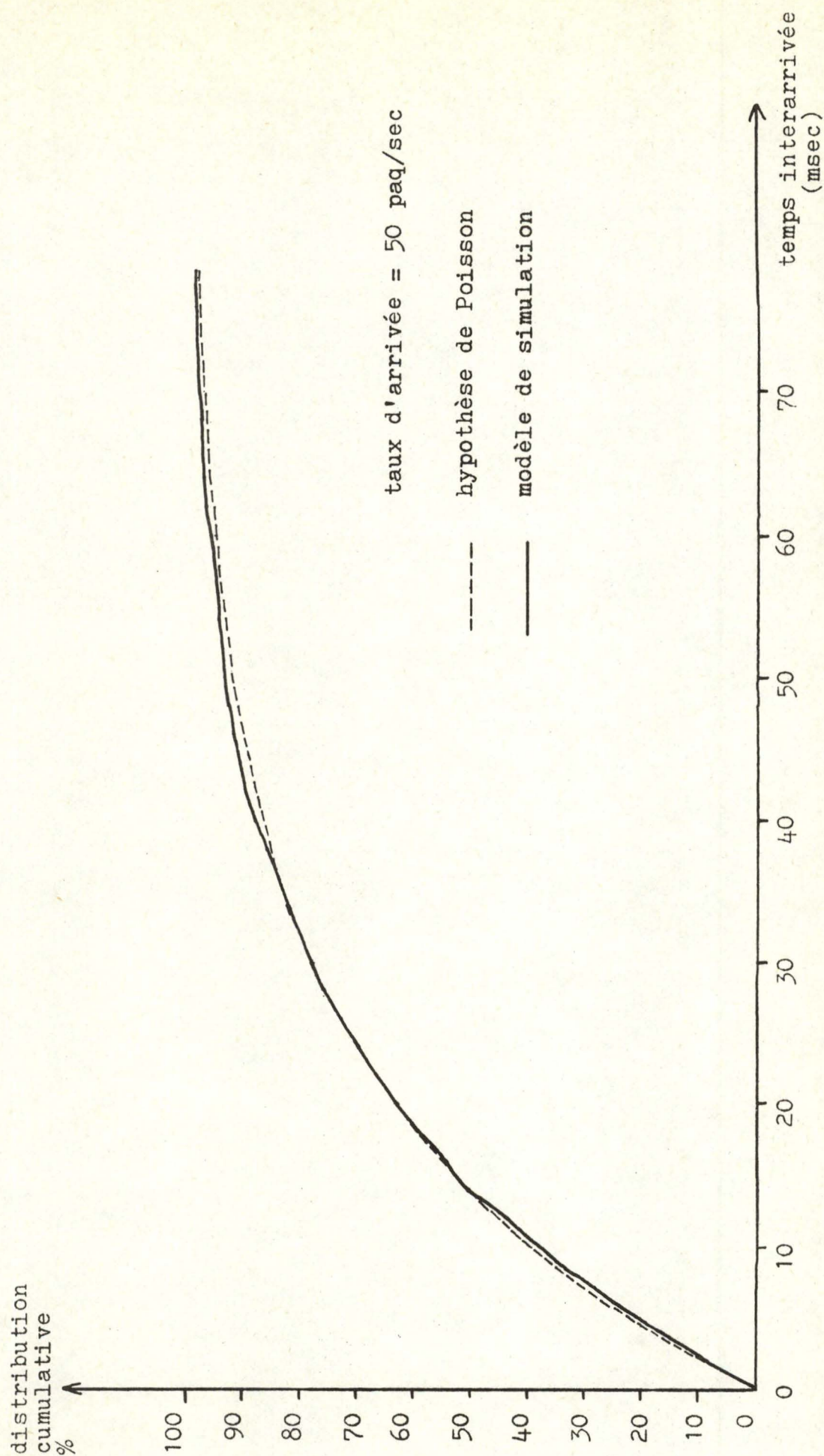


Fig. 4.4. Comparaison de la distribution du temps interarrivée obtenue avec l'hypothèse de Poisson et avec le générateur du modèle de simulation.



ce qui correspond à une longueur totale moyenne du paquet

$$\overline{LP} = \overline{LT}_{\text{tronq}} + h + s = 384.56 \text{ bits}$$

et à une durée d'émission théorique moyenne de 8.02 ms pour une capacité de ligne de 48 kb/s. La simulation nous a fourni une durée moyenne d'émission de 8.16 ms. La faible différence par rapport à la valeur théorique s'explique par le comportement du générateur.

b) la file d'attente de traitement.

Le système formé par la file d'attente IQ et le serveur de traitement à temps de service constant est un système d'attente M/D/1 car les arrivées suivent un processus de Poisson. Si  $\rho$  est le taux d'occupation du serveur et  $s$  le temps de service constant, le nombre moyen d'unités en file d'attente  $\bar{n}$  et le temps moyen d'attente en file  $\bar{w}$  s'expriment respectivement par:

$$\bar{n} = \frac{\rho^2}{2(1-\rho)} \quad \text{et} \quad \bar{w} = s \frac{\rho}{2(1-\rho)}$$

Pour obtenir le nombre moyen d'unités en file par la simulation, nous divisons la somme des durées d'attente en file par la durée totale de la simulation.

Le tableau 4.3. compare les résultats théoriques et les résultats obtenus par simulation pour  $\bar{n}$  et  $\bar{w}$ . ( $s = 5 \text{ ms}$ )

taux d'arrivée (paq/s)		30	50	70	100
taux d'occupation		0.15	0.25	0.35	0.50
$\bar{n}$	simulation	0.01	0.04	0.09	0.25
	théorique	0.01	0.04	0.09	0.25
$\bar{w}$	simulation	0.44	0.83	1.37	2.52
	théorique	0.44	0.83	1.35	2.50

Tableau 4.3



Nous voyons que la durée moyenne d'attente en IQ est toujours largement inférieure à la durée moyenne en OQ, même lorsque le taux d'utilisation du serveur de traitement vaut 0.5 (taux d'arrivée de 100 paquets par seconde).

Or nous avons pris des temps de traitement (routine "tâche-principale") et des temps d'exécution des routines modem-noeud plus important que ceux utilisés en pratique (dans Arpa, ces temps sont respectivement de 0.240 msec, 0.15 msec et 0.15 msec). D'autre part, on atteint jamais en pratique des taux d'arrivée de paquets à un noeud aussi élevés que 100 paq/sec. On peut affirmer que l'IQ n'influence que très légèrement le comportement dynamique des files de sortie.

c) La file des paquets en attente d'acquiescement (SQ).

Comme il n'y a pas de trafic inverse, la ligne du noeud 2 vers le noeud 1 est uniquement utilisée pour la transmission des paquets vides contenant les ACK.

Puisque le délai de retour de l'ACK est constant (il vaut le temps de transmission d'un paquet vide plus les temps d'exécution des routines noeud-modem et modem-noeud) et qu'il est inférieur au temps interarrivée minimum des paquets au noeud 2 (qui vaut le temps de transmission du paquet normal le plus court), il n'y aura jamais qu'un seul paquet en SQ. Ce n'est qu'en cas d'existence de trafic inverse que l'on peut étudier l'influence du comportement dynamique du nombre de paquet en SQ (cf. 4.2.)

d) La file de sortie.

Les paquets passant par la file d'attente de sortie n'arrivent pas dans la file suivant un processus de Poisson puisqu'ils proviennent d'un système M/D/1; le service (transmission du paquet) à une distribution "exponentielle + constante" (c'est à dire la distribution de la longueur des paquets). On ne peut donc pas comparer les résultats obtenus aux résultats théoriques d'un système M/M/1.



Le tableau 4.1. nous fournit la durée moyenne d'attente en OQ pour différents taux d'arrivée des paquets au noeud 1 et pour deux vitesses de ligne différentes. Pour un même taux d'arrivée de 30 paquets/seconde, la durée moyenne en file augmente d'un facteur très important (environ 20) si l'on diminue la vitesse de la ligne d'un facteur 2.5, c'est-à-dire si l'on augmente le taux d'occupation du serveur (la ligne) de ce même facteur, pour arriver à une valeur proche de 1. C'est un fait bien connu dans les systèmes d'attente du type M/M/1.

Lorsque le taux d'occupation de la ligne est supérieur à 1 (essai 4), on ne peut plus atteindre un régime stationnaire. La figure 4.3. montre l'évolution du nombre de paquets en OQ dans ce cas, (la ligne permet d'envoyer 96.13 paquets par seconde alors que le taux d'arrivée au noeud est de 100 paquets par seconde). On peut remarquer cependant que l'instabilité reste modérée puisqu'on constate un retour momentané du temps d'attente à zéro msec aux environs du 850<sup>ième</sup> paquet. La durée moyenne d'attente est évidemment très élevée, et il y a en moyenne plus de 13 paquets en attente.

Si le nombre de paquets en (OQ + SQ) est limité à 8 (comme dans Arpa); il faut arriver à un taux d'occupation de la ligne de plus de 0.70 pour atteindre ces 8 paquets s'il n'y a pas de trafic inverse.

Si l'on compare le nombre de paquets et le nombre de bits en OQ, on obtient respectivement les rapports suivants en fonction du taux d'arrivée:

Essais	1	2	3	5	4
taux d'arrivée	30	50	70	30	100
rapport $\frac{\text{nbre de bits}}{\text{nbre de paq.}}$ (nbres moyens).	1571	976	717	664	502

↑  
capacité ligne plus faible

Tableau 4.4.



Ces résultats montrent que moins le système est chargé, plus ce sont les paquets longs qui causent une attente en OQ. Si le système est très chargé (essai 4), on atteint un rapport 502 presque égal à la longueur moyenne des paquets (500 bits).

Si l'on compare de la même façon les nombres maximum de bits et de paquets, leur rapport varie de 776 à 561, ce qui indique que, pour un essai, lorsqu'on atteint le maximum, la longueur moyenne des paquets en file est supérieure à la longueur moyenne de tous les paquets: les paquets longs causent une attente plus longue. C'est un résultat auquel on pouvait s'attendre.

La figure 4.6. montre l'évolution dans le temps des nombres de paquets et de bits en file OQ, pour une tranche de temps de 1.7 sec et pour l'essai n° 2. Nous avons obtenus des graphiques semblables pour les autres essais. On constate que pour un nombre de paquets élevé, on obtient également un nombre de bits élevé. Mais d'autre part, pour seulement 2 paquets en file, on peut dépasser 2000 bits (ce qui correspond à deux paquets très longs; la longueur maximum des paquets est de 1176 bits). Les variations du nombre de bits sont donc beaucoup plus brutales.

Ces comparaisons montrent que si l'estimation du délai d'attente en file de sortie d'un noeud  $p$ , ( $W^p$ ) qui intervient dans la mise à jour de la table des délais du noeud, est mesurée en proportion du nombre de paquets en file d'attente, l'estimation sera beaucoup moins précise que si elle est mesurée en proportion du nombre de bits en attente; cette dernière estimation étant elle la meilleure puisque le nombre de bits est en rapport direct avec le temps qu'il faut pour transmettre les paquets correspondants.

La figure 4.6. bis donne l'évolution du nombre de paquets en OQ pour l'essai 1.



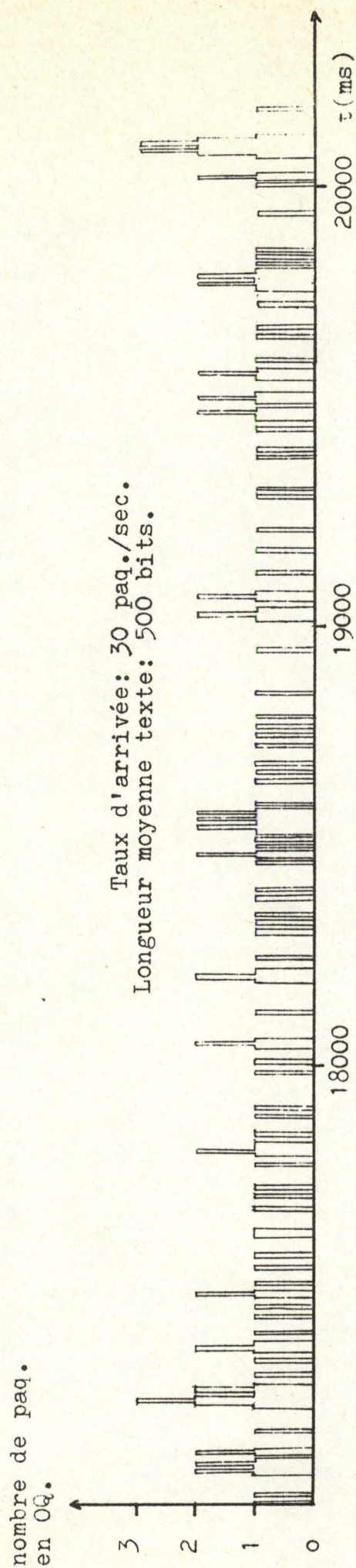


Fig. 4.6. bis

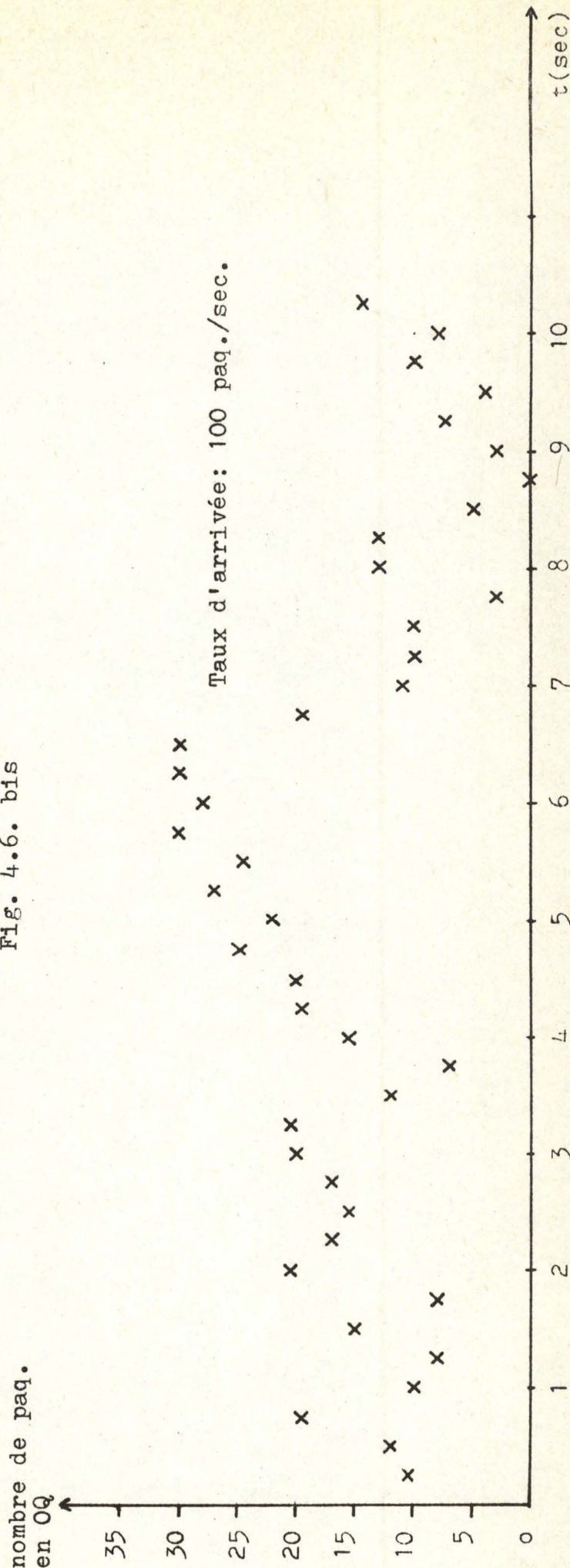


Fig. 4.3. Evolution du nombre de paquets en OQ (essai 4)



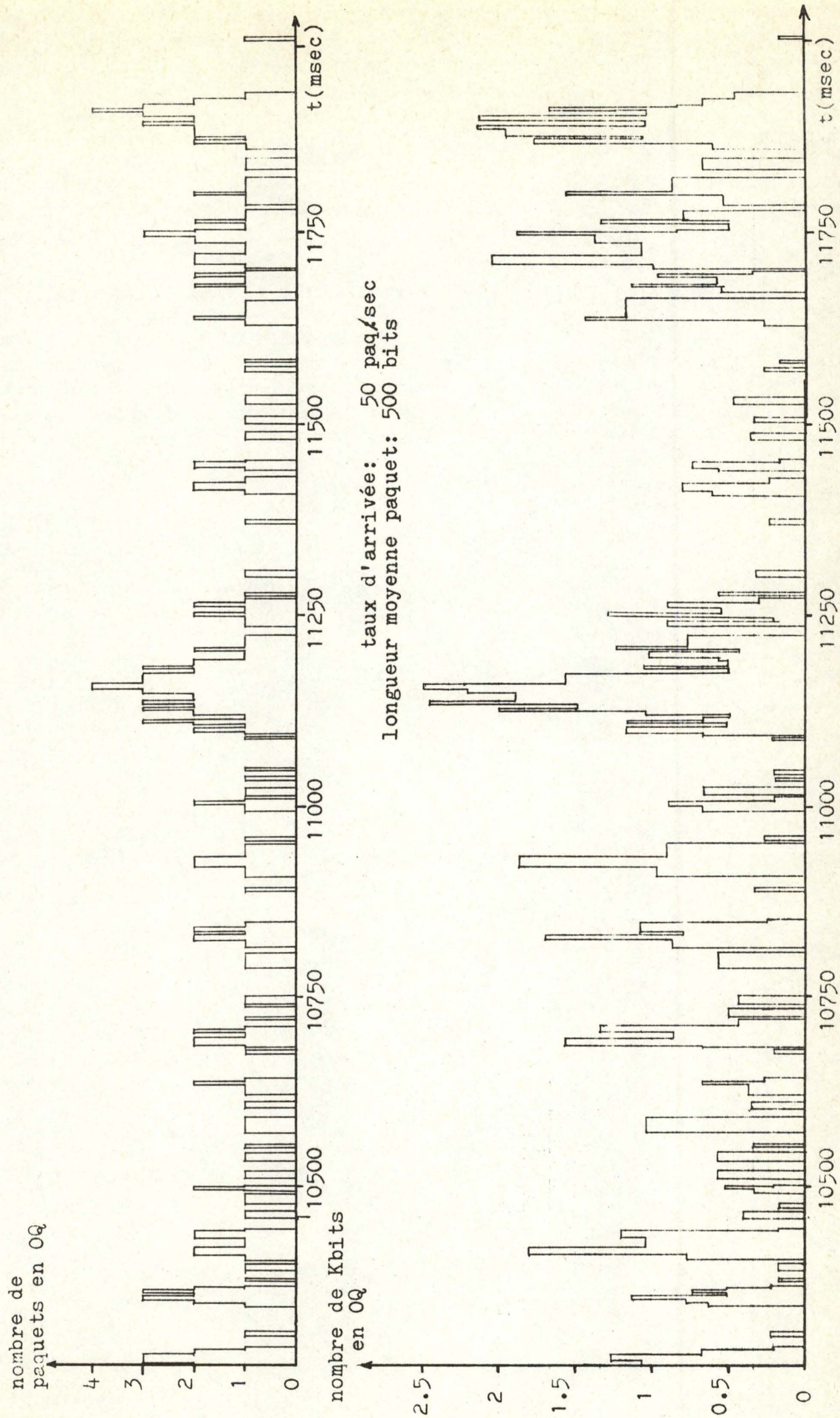


Fig.4.6 comparaison du nombre de paquets et du nombre de bits en OQ sur la même période de temps et dans les mêmes conditions



### 4.3. Simulation entre 2 noeuds avec trafic inverse.

#### 4.3.1. Le programme de simulation.

Le programme étudie le comportement des file d'attente des 2 noeuds.

Il comporte 3 parties:

- a) un générateur de paquets simulant l'arrivée des paquets dans les files d'entrée de chaque noeud.
- b) un programme de simulation, suivant le protocole C pour la routine noeud-modem, exécuté suivant diverses conditions aux 2 noeuds.
- c) un ensemble de sortie permettant l'analyse macroscopique et statistique des résultats.

Le système simulé est illustré à la figure 4.8.

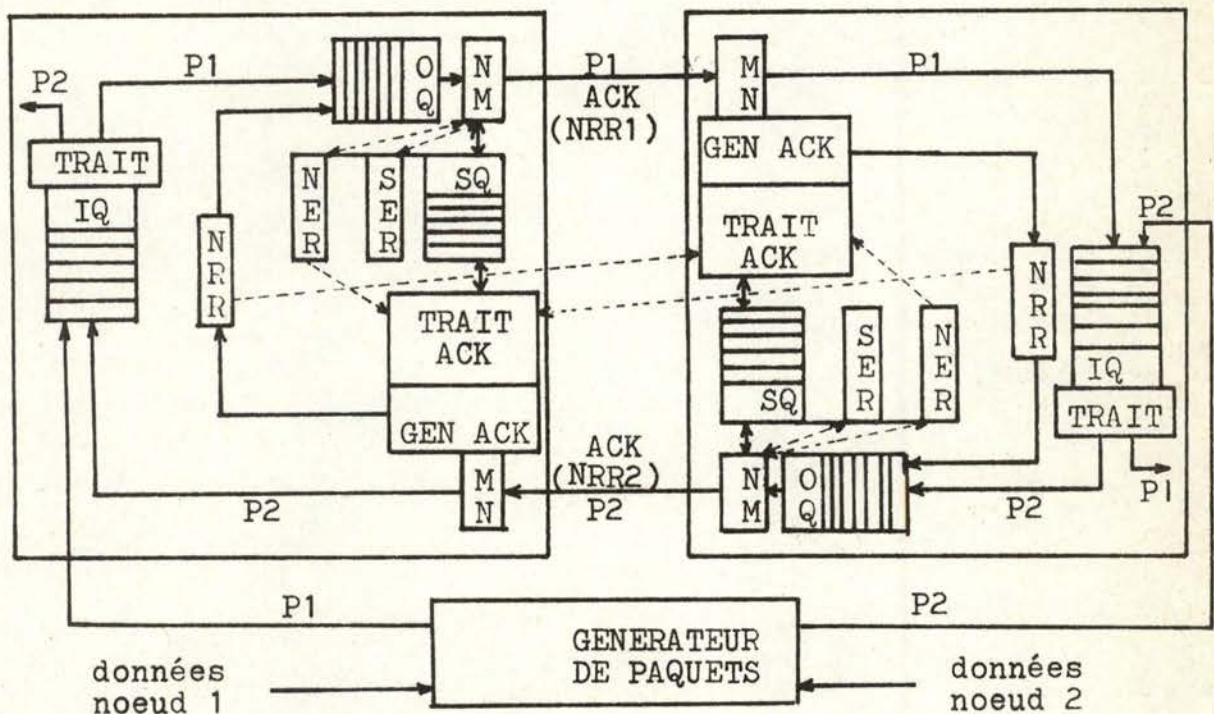


Fig. 4.8.



### Le générateur de paquets.

Pour chaque noeud, le générateur construit une table contenant l'heure d'arrivée  $t_1$  des paquets (heure d'interruption de la routine modem-noeud) et une table contenant la longueur totale de chaque paquet; de la même façon que pour le programme du paragraphe 4.2.

On supposera que tous les paquets arrivant à un noeud et provenant du générateur sont destinés à la file OQ étudiée. Les arrivées en chaque noeud suivant un processus de Poisson de taux  $k$  (paquets/seconde), le temps interarrivée a donc une distribution exponentielle de moyenne  $1/k$  (seconde). La longueur totale  $LP$  d'un paquet est la somme d'une longueur constante et d'une longueur variable (qui suit une distribution exponentielle tronquée).

$$LP = k + s + LT$$

$k + s$  : constante

$LT$  : variable de distribution exponentielle tronquée en  $LT_{\max}$

### Les temps significatifs.

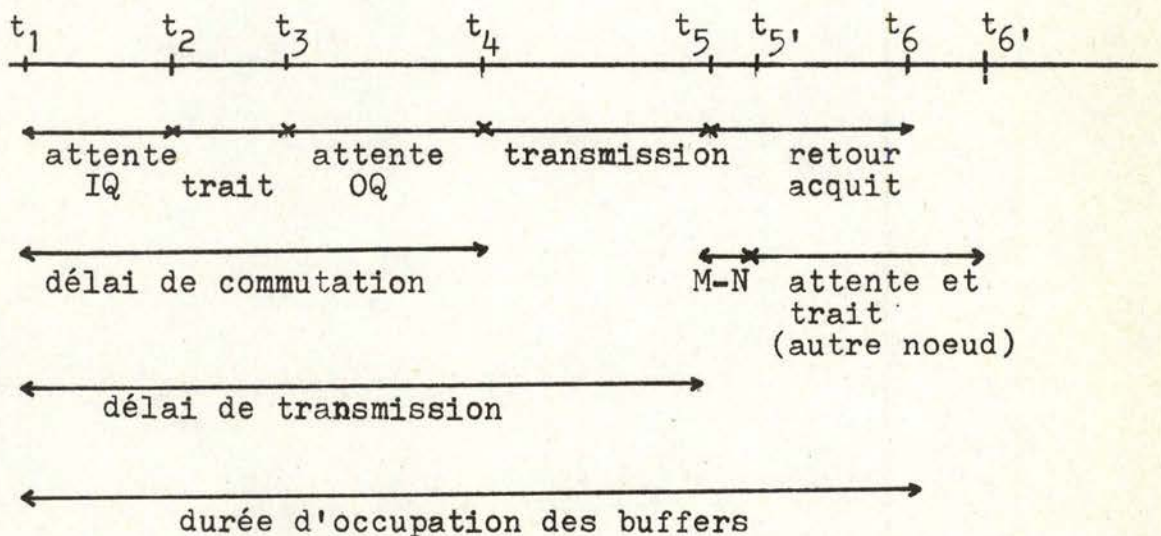


Fig. 4.9.



Les conditions de test.

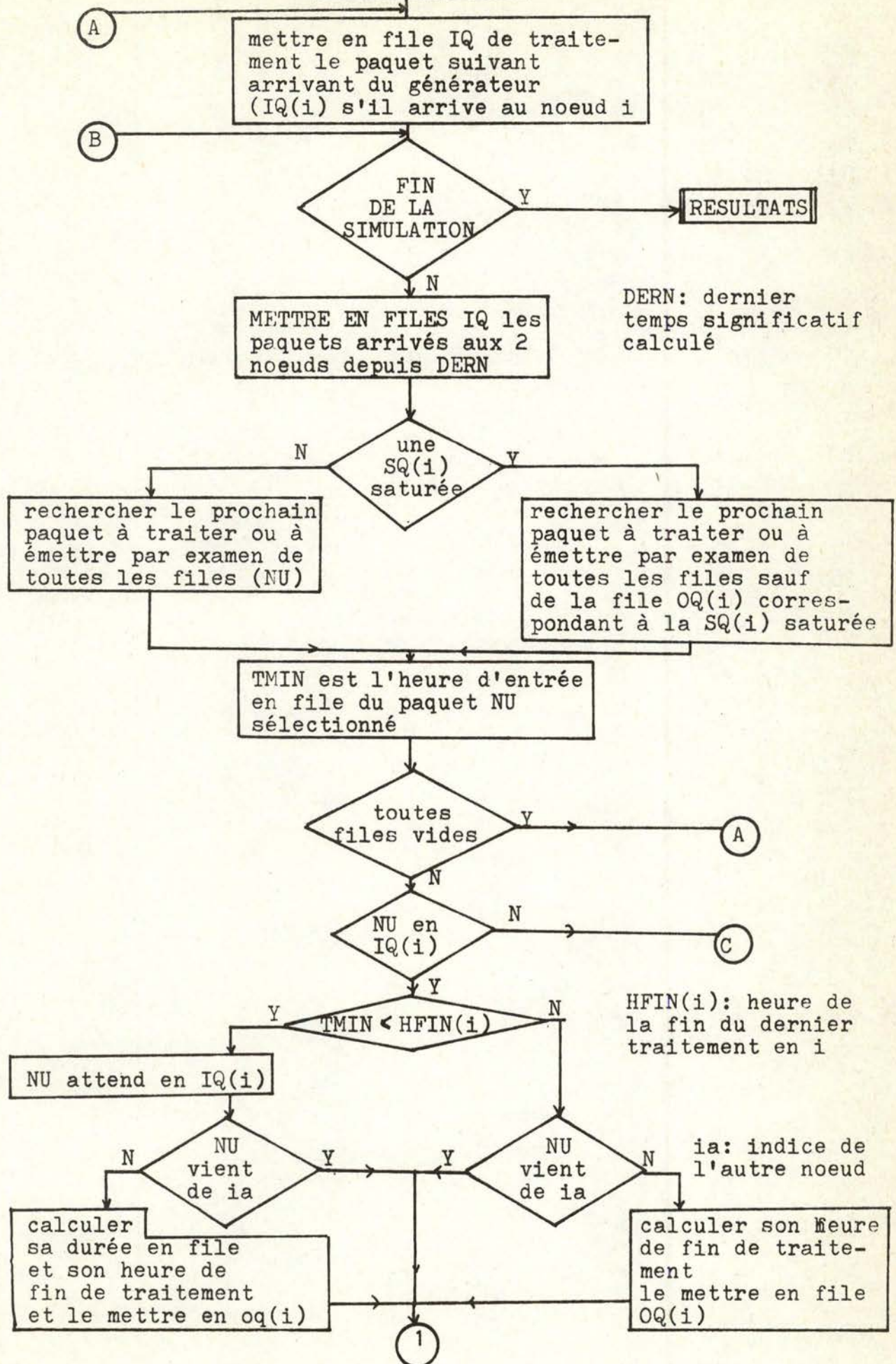
Les divers essais ont été réalisés sous les conditions:

- 1) la capacité en buffer n'est pas limitée, sauf bien sûr pour la SQ qui est limitée à 8.
- 2) chaque paquet est reçu correctement.
- 3) pour chaque paquet arrivant au noeud 1, on a adopté la même longueur moyenne de texte  $\overline{LT} = 512$  bits (moyenne de la distribution non tronquée) et le même taux d'arrivée de 50 paquets/seconde. Au noeud 2, la longueur moyenne du texte et le taux d'arrivée varient d'un essai à l'autre.
- 4)  $h = 72$  bits;  $s = 80$  bits;  $LT_{\max} = 1024$  bits.
- 5) la capacité de la ligne est 48 kbits/sec; le temps d'exécution de la routine noeud-modem et de la routine modem-noeud est de 0.5 msec (ou 0.1 msec); le temps d'exécution de la routine "tâche principale" (traitement d'un paquet) est de 4 msec (ou 0.3 msec).
- 6) le délai de propagation sur la ligne est négligeable.
- 7) la simulation porte sur 1000 paquets (sauf lorsque les taux d'arrivée aux 2 noeuds sont différents).

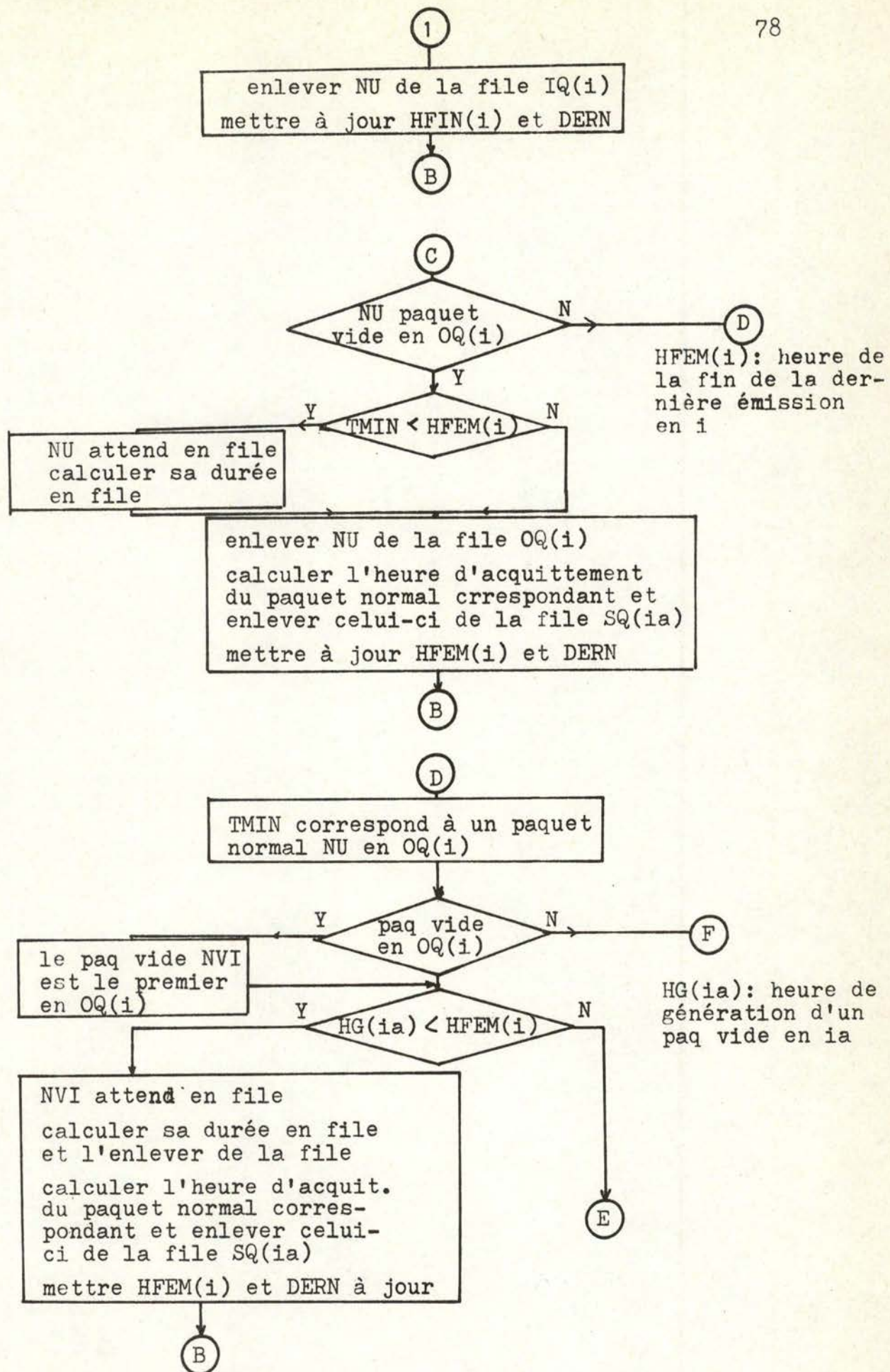
A partir d'une exécution de la simulation, on peut obtenir les différents temps significatifs  $t_2$  à  $t_6$  et les délais montrés à la figure 4.9., pour chaque paquet; ainsi que les grandeurs statistiques habituelles. On obtient aussi des résultats sur les longueurs des diverses files IQ, OQ, SQ, en nombre de paquets et nombre de bits.

Un organigramme retrace l'évolution du programme de simulation dans ses grandes lignes dans les 4 pages qui suivent.

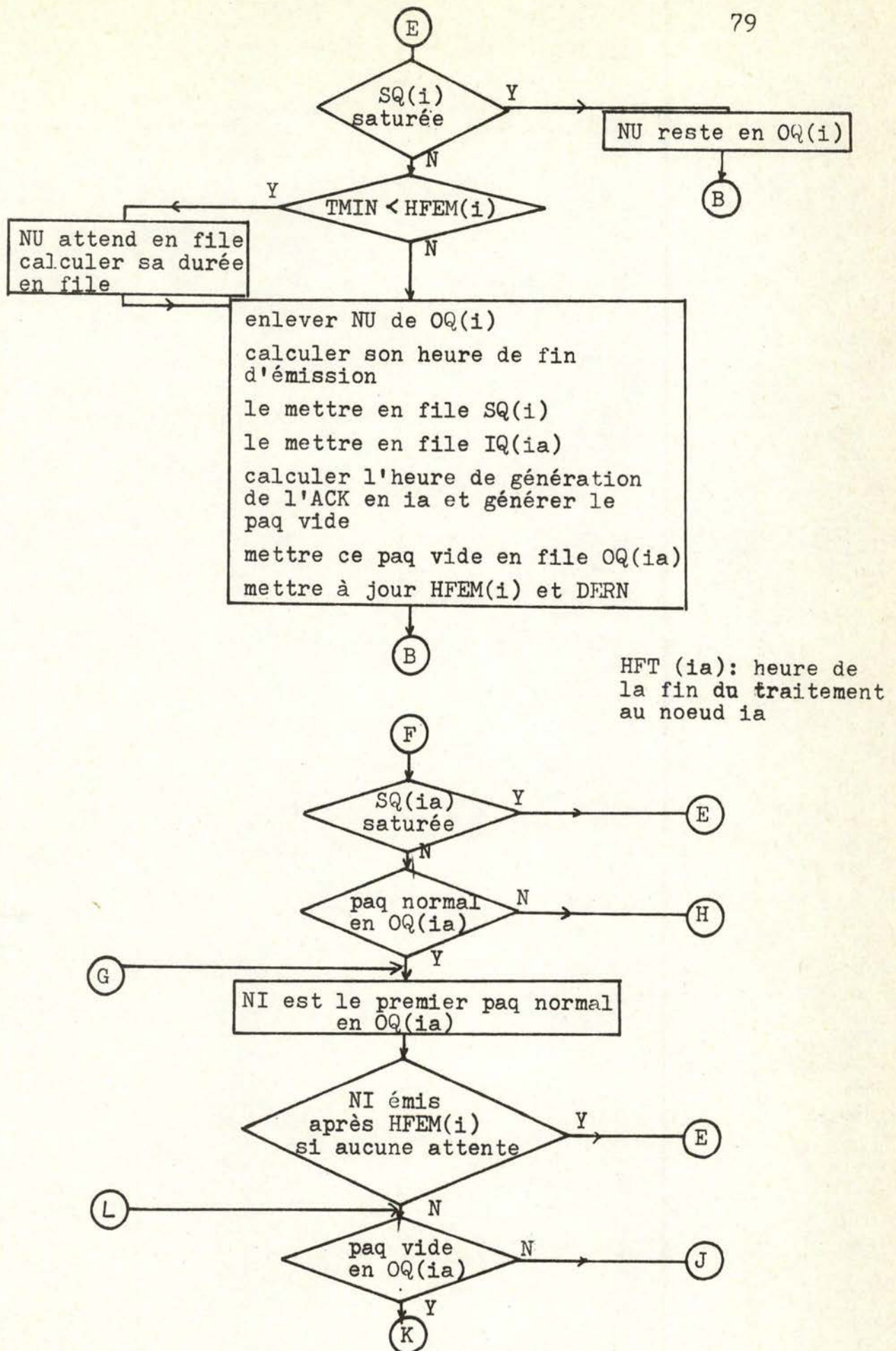




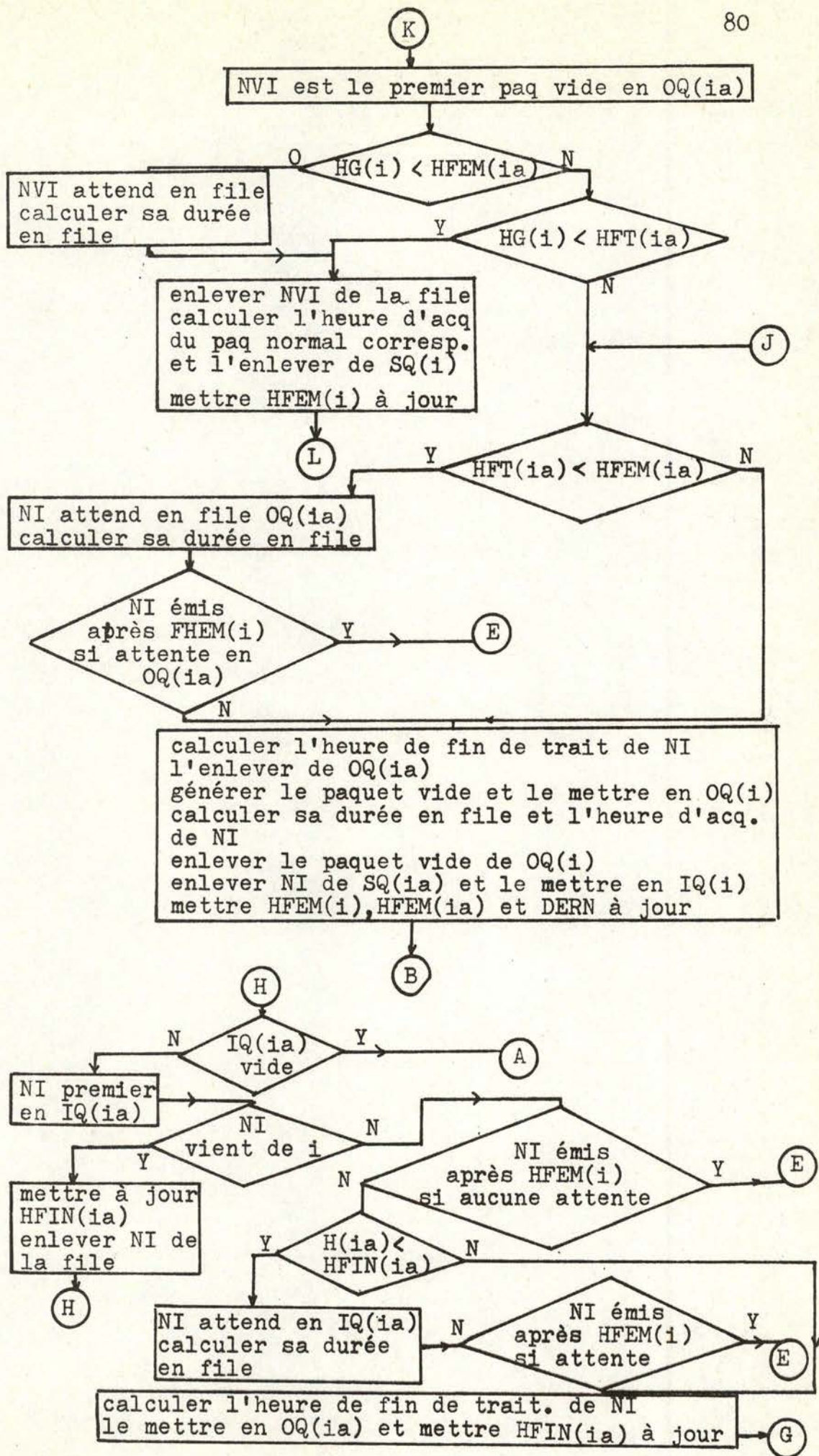














ESSAIS	1		2		3		4		5	
	NOEUD1	NOEUD2	NOEUD1	NOEUD2	NOEUD1	NOEUD2	NOEUD1	NOEUD2	NOEUD1	NOEUD2
taux d'arrivée (paq/s)	50	50	50	50	50	50	50	100	50	50
longueur moyenne du texte (bits)	512	512	512	128	512	256	512	256	512	256
temps d'exécution M-N et N-M	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.1	0.1
temps de traitement (ms)	4.0	4.0	4.0	4.0	4.0	4.0	4.0	4.0	0.3	0.3
durée moyenne d'attente en IQ (ms):	1.08	1.13	1.16	1.12	1.17	1.09	1.96	2.56	0.005	0.004
durée d'attente en OQ (ms)										
moyenne	7.14	8.16	7.08	1.30	7.01	4.82	9.81	42.00	7.65	5.55
maximum	70.4	90.8	73.0	18.4	71.6	57.5	69.6	140.5	74.0	68.3
durée moyenne de transmission (ms)	10.40	10.40	10.40	5.81	10.40	8.17	10.40	8.17	10.40	8.17
durée d'attente du PV dans l'autre noeud										
moyenne	3.88	3.75	1.091	3.34	2.45	3.48	4.51	3.97	2.42	3.46
maximum	22.6	22.1	12.1	21.0	21.8	22.8	23.0	23.0	22.8	22.5
durée de retour de l'acquit (ms)										
moyenne	6.55	6.41	3.76	6.01	5.12	6.15	7.17	6.64	4.28	5.32
maximum	25.3	24.8	14.8	23.7	24.5	25.5	25.7	25.7	24.5	24.2
charge totale (kb/s)	28.0	28.6	28.0	17.7	28.0	23.2	31.8	42.5	28.0	23.2
en % de la capacité de la ligne	58.3	59.7	58.3	36.9	58.3	48.4	66.4	88.7	58.3	48.4

Tableau 4.6. Résultats aux noeuds 1 et 2: délais.



ESSAIS	1		2		3		4		5	
	NOEUD1	NOEUD2	NOEUD1	NOEUD2	NOEUD1	NOEUD2	NOEUD1	NOEUD2	NOEUD1	NOEUD2
taux d'arrivée (paq/s)	50	50	50	50	50	50	50	100	50	50
longueur moyenne du texte (bits)	512	512	512	128	512	256	512	256	512	256
temps d'exécution M-N et N-M	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.1	0.1
temps de traitement (ms)	4.0	4.0	4.0	4.0	4.0	4.0	4.0	4.0	0.3	0.3
nombre de paquets en OQ										
moyenne (PN+PV)	0.53	0.59	0.50	0.12	0.51	0.36	0.67	2.30	0.54	0.39
maximum (PN)	6	7	6	4	6	6	6	17	6	7
nombre max. de bits en OQ	3456	5406	3456	1387	3385	2336	3456	6816	4163	3600
nombre maximum de paquets en SQ	3	4	3	4	4	5	4	5	4	5
nombre maximum de paquets en (OQ+SQ)	7	7	7	5	7	8	7	19	7	8

Tableau 4.7. Résultats aux noeuds 1 et 2: files d'attente.



#### 4.3.2. Analyse des résultats.

Les tableaux 4.6 et 4.7 donnent les résultats principaux obtenus dans 5 essais, ainsi que les valeurs des paramètres pour ces essais.

Le tableau 4.6 se rapporte aux divers délais mesurés.

Le tableau 4.7 se rapporte aux nombres d'éléments (bits et paquets dans les files d'attente).

Nous avons parlé du comportement du générateur et de l'effet d'une troncature sur la distribution de la longueur du texte d'un paquet au paragraphe 4.4.4., nous n'y reviendrons pas.

Les résultats obtenus pour un noeud seront surtout analysés en fonction du trafic inverse et des conditions dans l'autre noeud.

Les trois premiers essais sont effectués avec les mêmes conditions au noeud 1 et avec des longueurs moyennes de paquets différentes au noeud 2.

Le 4e essai est semblable au 3e pour les longueurs moyennes des paquets au noeud 2 mais le taux d'arrivée à ce noeud est doublé.

Le 5e essai est semblable au 3e pour les longueurs moyennes de paquets et les taux d'arrivée mais les divers temps d'exécution (M-N, N-M et traitement) ont été considérablement réduits (valeurs du même ordre de grandeur que celles utilisées par Arpa).

Nous n'avons pas repris dans le tableau 4.6. les délais de commutation, de transmission et de durée d'occupation des buffers (ils s'obtiennent facilement par addition de différentes durées, comme on le voit à la figure 4.9.)

Nous y avons inclus la charge totale sur la ligne (due aux paquets normaux et aux paquets vides) et le pourcentage de la capacité de la ligne qu'elle représente).



Par rapport au programme du paragraphe 4.2. où il n'y avait pas de trafic inverse, la durée en IQ est plus élevée.

Cela est dû aux paquets du trafic inverse qui arrivent à un noeud et y sont en compétition pour le traitement avec les paquets arrivant du générateur.

Le processus d'arrivée des paquets au serveur de traitement n'est plus un processus de Poisson; les résultats obtenus ne peuvent plus être comparés à ceux obtenus pour un système d'attente M/D/1.

Dans les trois premiers essais, on obtient, au noeud 1, des temps moyens d'attente en IQ de 1.08, 1.17, 1.16 pour des longueurs de paquets du trafic inverse de 512, 256 et 128 bits respectivement. Au noeud 2, on obtient 1.13, 1.09, 1.12 pour le même trafic inverse. Donc, dans les mêmes conditions de taux d'arrivée aux 2 noeuds, les variations de la durée d'attente en IQ sont très faibles.

L'essai 4 est plus intéressant: pour le noeud 1, le trafic inverse se compose de paquets deux fois plus fréquents et plus courts: l'attente en IQ des paquets provenant du générateur est pratiquement doublée par rapport à l'essai 3 et peut influencer plus l'attente au serveur suivant (OQ). Pour le noeud 2, le trafic inverse se compose de paquets deux fois moins fréquents mais plus longs: l'attente en IQ des paquets provenant du générateur est multipliée par un facteur 2.5. par rapport à l'essai 3 où le taux d'arrivée est deux fois plus petit. (Notons que l'effet de troncature sur la longueur des paquets fournit au noeud 2 des paquets de longueur supérieure à la moitié de la longueur des paquets au noeud 1).

La durée d'attente en IQ au noeud 2 n'influence pratiquement pas la durée d'attente en OQ car la charge totale de la ligne dans la direction 2→1 est très élevée (88% de la capacité de la ligne).



Enfin pour l'essai 4, où nous avons adopté des temps d'exécution M-N, N-M et de traitement très petit (0.1, 0.1 et 0.3 respectivement), la durée moyenne d'attente est pratiquement nulle et donc sans influence sur le serveur suivant. On peut dans ce cas affirmer que le processus d'arrivée des paquets devant la ligne est un processus de Poisson de même taux que le processus d'arrivée devant le serveur de traitement.

En conclusion, on peut dire:

- une dissymétrie complète (en fréquence des paquets arrivant et en longueur moyenne de texte), comme dans l'essai 4, influence le temps d'attente dans les files IQ des 2 noeuds (augmentation au moins double par rapport à l'essai 3).
- une forte dissymétrie de la longueur moyenne des paquets pour des taux d'arrivée égaux influence fortement l'attente en OQ au noeud 2 (essai 2) puisque dans ce cas, les paquets arrivent en moyenne et au minimum dans l'OQ du noeud toutes les 5.12 ms (durée moyenne d'attente en IQ + durée du traitement) alors que le temps de service moyen du serveur (ligne) vaut 5.81 ms pour la longueur moyenne des paquets au noeud 2 ( $127 + 152 = 289$  bits).

Ce qui a pour effet de donner un temps d'attente faible en OQ (1.30 ms). (ce temps d'attente dépend aussi des PV envoyés sur la ligne dans la même direction comme nous le verrons en analysant les files de sortie).



b) La file SQ des paquets en attente d'acquittement.

Dans tous les essais effectués, le nombre de paquets en attente d'acquittement varie de 3 à 4 au noeud 1 et de 4 à 5 au noeud 2 (cf tableau 4.7).

On n'atteint donc jamais la saturation des files SQ, même lorsque les conditions de charge sont élevées; cela s'explique par le fait qu'en utilisant le protocole C pour la routine noeud-modem, on libère au plus tôt les buffers de la SQ.

On peut néanmoins constater que l'attente moyenne d'un paquet dans les SQ des 2 noeuds varie d'une façon non négligeable suivant les conditions de chaque essai.

Pour les essais 1 à 3 (même taux d'arrivée aux 2 noeuds), on observe le comportement suivant:

- au noeud 1, un paquet reste respectivement 6.55, 5.12 et 3.76 ms en SQ pour une longueur moyenne des textes de paquet de 512, 256 et 128 bits au noeud 2: un paquet normal étant plus vite transmis dans la direction 2→1, la ligne est plus souvent disponible pour le renvoi des ACK. Dans le cas où les paquets normaux ont une petite longueur (essai 3), un paquet vide ne passe jamais plus de 12.1 ms dans la file OQ du noeud 2.
- au noeud 2, les durées en SQ, sont pratiquement les mêmes (6.41, 6.15 et 6.04 ms respectivement). Les PN arrivant dans l'OQ du noeud 1 sont toujours de même longueur moyenne; les PV y arrivant y attendent pratiquement le même temps (puisque les PN qu'ils acquittent arrivent au noeud 1 à la même fréquence, mais avec une occupation moindre de la ligne 2→1 s'ils ont une longueur moyenne plus petite).

Dans l'essai 4, l'effet de dissymétrie des taux d'arrivée amène des durées d'attente dans les SQ des 2 noeuds supérieures à celles obtenues dans le cas où il y a symétrie des taux



d'arrivée (essai 3).

La différence est surtout marquée au noeud qui a le taux d'arrivée le plus faible, le noeud 1: en effet, les paquets normaux arrivant dans l'OQ du noeud 2 à une fréquence double, les paquets vides doivent y attendre plus longtemps (4.51 ms contre 2.45 ms à l'essai 3).

La différence au noeud 2 s'explique par le fait qu'il y a deux fois plus de PV générés au noeud 1 et destinés à acquitter des paquets normaux venant du noeud 2. Ces PV devront être en files plus longtemps parce qu'ils sont plus nombreux. La différence avec l'essai 3 (3.97 ms contre 3.48 ms) est assez faible; mais il ne faut pas oublier que les PV sont prioritaires par rapport aux PN pour la transmission sur une ligne.

Enfin dans l'essai 5, où les temps d'exécution des routines ont été réduit par rapport à l'essai 3, les temps d'attente des paquets vides sont pratiquement identiques dans les 2 essais (2.42 contre 2.45 dans la file OQ du noeud 2 et 3.46 contre 3.48 dans la file OQ du noeud 1); les différences dans les durées moyennes d'attente en SQ résultent donc seulement de la différence des temps d'exécution des routines N-M et M-N (puisque  $\text{attente SQ} = \text{tps d'exec de M-N} + \text{attente du PV} + \text{tps d'exec N-M}$ ).

En conclusion, on peut affirmer que le trafic inverse influence la durée en SQ. Celle-ci dépend aussi des temps d'exécution des routines (mais il est évident que dans un réseau, ces temps d'exécution sont en général fixés une fois pour toutes).

En raison de la priorité de la transmission des paquets vides, on atteint jamais une durée d'attente en SQ très importante (le maximum obtenu dans les divers essais est de 25.7 ms à l'essai 4, lorsque la charge totale sur la ligne est la plus grande).



c) La file de sortie OQ.

Dans ce système, le comportement de la file OQ d'un noeud est influencé par les conditions du trafic inverse, par l'intermédiaire des paquets vides PV générés au noeud pour acquitter des paquets normaux PN venant de l'autre noeud. Ces paquets vides sont en compétition avec les paquets normaux pour la transmission vers l'autre noeud, ils sont prioritaires, mais le service en cours ne peut pas être interrompu.

Le tableau 4.6. nous fournit les durées moyennes et maximum d'attente des PN dans les files OQ des noeuds.

Nous avons déjà parlé de la durée moyenne des PV en OQ dans le point b).

Dans les 3 premiers essais (taux d'arrivée aux deux noeuds identiques), on observe les comportements suivants:

- au noeud 1, la durée moyenne en OQ vaut respectivement 7.14, 7.08 et 7.01, c'est-à-dire qu'elle ne varie pratiquement pas; ce qui est normal puisque les paquets normaux ont la même longueur moyenne et que les paquets vides en compétition arrivent à la même fréquence.
- au noeud 2, la durée moyenne en OQ des PN varie plus que proportionnellement à leur longueur moyenne. Cette variation est due à la durée moyenne de transmission (proportionnelle à la longueur moyenne des paquets), mais aussi au rapport entre le taux d'arrivée à la file OQ (pratiquement identique pour les 3 essais) et le taux de service de la ligne.

Dans l'essai 4, où le taux d'arrivée au noeud 2 est doublé par rapport à celui de l'essai 3, les durées d'attente sont augmentées.



Au noeud 1, l'augmentation est modérée (on obtient 9.81 ms contre 7.01 ms à l'essai 3). Cela s'explique par la fréquence des PV arrivant dans l'OQ, qui a été doublée et par la priorité de transmission des PV. Ces deux facteurs font augmenter la durée moyenne des PN en OQ. On constate d'ailleurs que le taux d'utilisation de la ligne 1→2 passe de 28.0 à 31.8 kb/s, puisqu'il y a plus de PV à transmettre.

Au noeud 2, l'augmentation est très importante (on obtient 42.00 ms contre 4.82 ms à l'essai 3) puisque le fait de doubler le taux d'arrivée des paquets normaux dans l'OQ amène une charge totale égale à 88.7 % de la capacité du serveur (ligne). C'est un résultat auquel on pouvait s'attendre en examinant les systèmes d'attente théoriques lorsque le taux d'occupation du serveur est proche de 1.

Dans l'essai 5, où les temps d'exécution des routines ont été réduits, les durées d'attente en OQ sont légèrement augmentées par rapport à ceux de l'essai 3. Cela s'explique par le fait que, l'attente étant pratiquement nulle au serveur précédent (traitement), le temps interarrivée minimum au serveur (ligne) est diminué: il vaut le temps de traitement.

On constate enfin que la valeur maximum du temps d'attente varie entre 18.4 et 140.5 ms, valeurs correspondant à des charges totales de 36.9 et 88.7 kb/s sur une ligne. Ces valeurs maximales peuvent influencer le choix du seuil SC au delà duquel on peut considérer un noeud comme étant inaccessible, lors de la sélection d'une route dans l'algorithme de routage (cf paragraphe 3.6). D'après nos essais, qui se rapportent à des charges totales relativement élevées (de 36.7 % à 88.7 % de la capacité de la ligne), si on fixe le seuil SC à 100 ms, seul le temps d'attente au noeud 2 dans l'essai 4 dépasse à plusieurs reprises ce seuil. Si on le fixe à 75 ms, il sera dépassé au noeud 2 dans les essais 1 et 4).



Pour des charges totales plus faibles, ce seuil ne sera pratiquement jamais dépassé. Ceci est valable pour le temps d'accès à un noeud voisin; lorsqu'on s'intéresse au délai pour atteindre un noeud plus éloigné, il faut tenir compte du nombre moyen de noeud sur les routes entre les deux noeuds pour fixer le seuil SC.

Intéressons nous maintenant au nombre de paquets et au nombre de bits dans les files de sortie OQ.

La figure 4.10 compare l'évolution, dans la même tranche de temps, du nombre de paquets (PN + PV) dans l'OQ du noeud 1 pour les essais 1 et 2; le premier graphique de la figure 4.11 montre cette évolution pour l'essai 3. Dans les trois cas, le nombre maximum de paquets dans l'OQ sur toute la durée de simulation est de 6 et le nombre maximum de bits dans l'OQ est d'environ 3456. On voit, dans les graphiques, que si les pointes se situent aux mêmes instants, elles ne sont pas de même intensité à un instant fixé; elles sont tantôt plus importantes dans un essai, tantôt plus importantes dans un autre. Cela s'explique par le fait que les PV n'arrivent pas aux mêmes instants dans les 3 cas (bien qu'il en arrive le même nombre sur une période très longue); mais, ces PV étant transmis très rapidement (1.67 ms), l'allure générale dans le temps est semblable dans les 3 cas.

Si l'on revient au tableau de 4.7., on constate que le nombre maximum de paquets en OQ au noeud 1 vaut 6 dans tous les essais. Dans les essais 1 à 3, le nombre moyen de paquets en file d'attente décroît très lentement (0.53, 0.51 et 0.50) avec la longueur des paquets normaux venant de l'autre noeud. Dans l'essai 4, il est plus important (0.67) puisqu'il y a deux fois plus d'ACK à renvoyer à l'autre noeud (lorsqu'on compte en nombre de paquets, un paquet vide à le même poids qu'un paquet normal). Au noeud 2, le nombre moyen de paquets (PN + PV) diminue si la longueur moyenne des paquets diminue, ce qui semble normal. Le tableau 4.8 donne les rapports du nombre moyen de paquets et du nombre moyen de bits en OQ (avec et sans les PV):



ESSAI	1	2	3
Longueur moyenne du paquet (bis)	500	392	269
Nombre moyen de paquets en OQ (PN + PV)	0.59	0.36	0.10
Nombre moyen de bits en OQ (PN + PV)	288.96	174.21	57.38
Rapport $\frac{\text{nombre de bits}}{\text{nombre de paq.}}$ (PN + PV)	489.76	483.92	478.17
Rapport $\frac{\text{nombre de bits}}{\text{nombre de paq.}}$ (PN)	899.52	887.84	876.34

Tableau 4.8.

où le nombre moyen de bits en OQ (PN + PV) s'obtient par:

$$\text{nombre moyen de bits} = \frac{(\text{DOQ} + \text{DOV}) \times C}{2}$$

où DOQ = durée moyenne d'attente des PN en OQ

DOV = durée moyenne d'attente des PV en OQ

C = capacité de la ligne

et où le rapport RN du nombre moyen de bits en OQ et du nombre moyen de paquets en OQ pour les PN s'obtient par:

$$\text{RN} = 2 \times \text{RNV} - 80$$

$$\text{où RNV} = \frac{\text{nombre de bits en OQ (PN + PV)}}{\text{nombre de paq.}}$$

Ceci nous montre que:

- ce sont surtout les paquets longs qui causent une attente en file.
- plus la longueur moyenne des paquets est petite, plus les paquets très longs par rapport à la longueur moyenne causent une attente en file.



Dans l'essai 4, on atteint 17 paquets (ligne utilisée à 88.7%) et 6816 bits au maximum. Ce qui correspond à une moyenne de 401 bits en file d'attente (la moyenne totale de la longueur des paquets vaut 392), ce qui montre que lorsqu'on atteint un très grand nombre de paquets en file, ceux-ci ont une longueur moyenne très proche de la longueur moyenne du total des paquets.

Ces comparaisons entre nombre de bits et de paquets en OQ nous montre encore une fois que l'estimation du délai en file d'attente est beaucoup moins précise si elle est exprimée en fonction du nombre de paquets que si elle est exprimée en fonction du nombre de bits.

Si l'on observe enfin le nombre de paquets normaux en (OQ+SQ), il ne dépasse 8 que dans un cas: lorsque le pourcentage d'utilisation de la ligne 2→1 vaut 88.7 (essai 4).

Même dans ce cas, le maximum n'intervient pas lorsque la SQ est saturée, puisqu'à ce moment, il n'y a que 2 paquets dans la SQ. C'est donc le nombre de paquets en OQ qui détermine principalement le maximum.

Dans tous les autres essais, où la charge totale est inférieure à 60% de la capacité de la ligne, on atteint au plus 8 paquets en (OQ+SQ), ce qui correspond au maximum fixé dans le réseau Arpa.

L'utilisation du protocole C présente l'avantage important de libérer au plus tôt les buffers de la SQ; même dans des conditions de charges élevées, on atteint pas la saturation de la SQ, seule file dont le nombre maximum de buffers est obligatoirement 8.

Des nombres importants de paquets (et de bits) en file n'apparaissent que dans l'OQ.



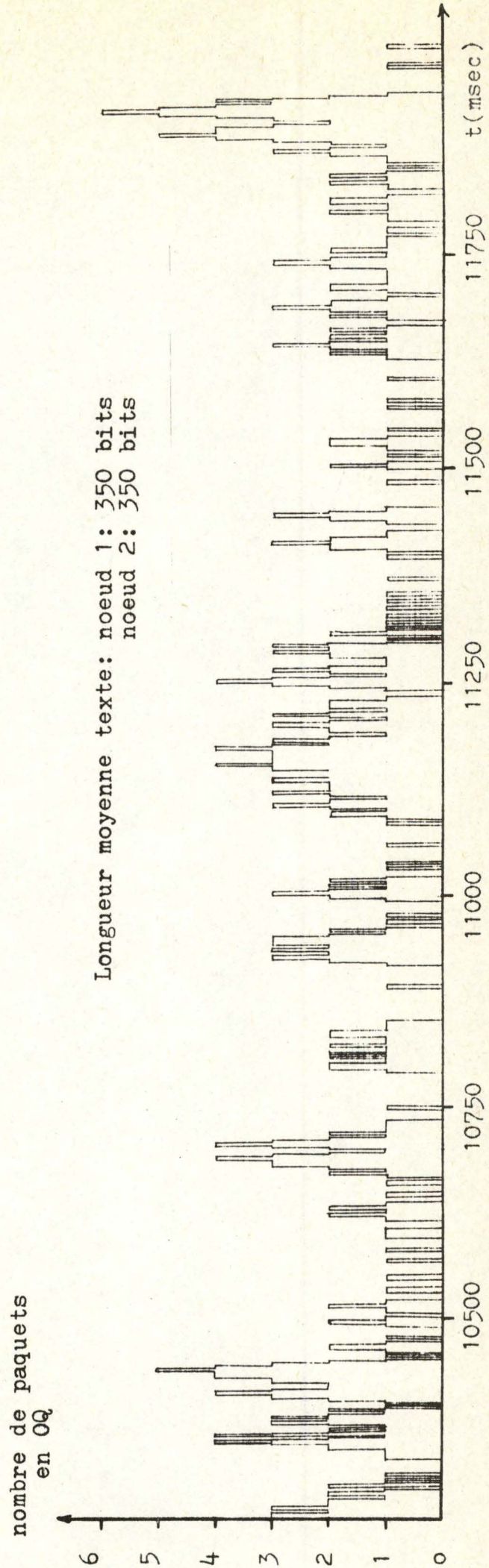
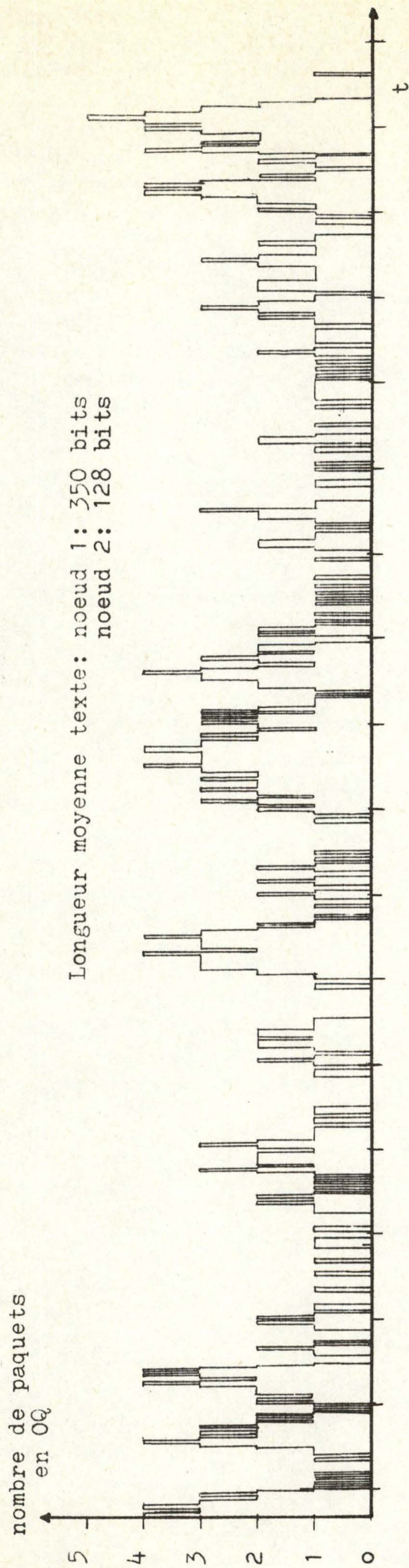


Fig. 4.10. Evolution du nombre de paquets en OQ au noeud 1 pour des longueurs de paquets différentes dans le trafic inverse (même taux d'arrivée aux 2 noeuds, même période de temps).



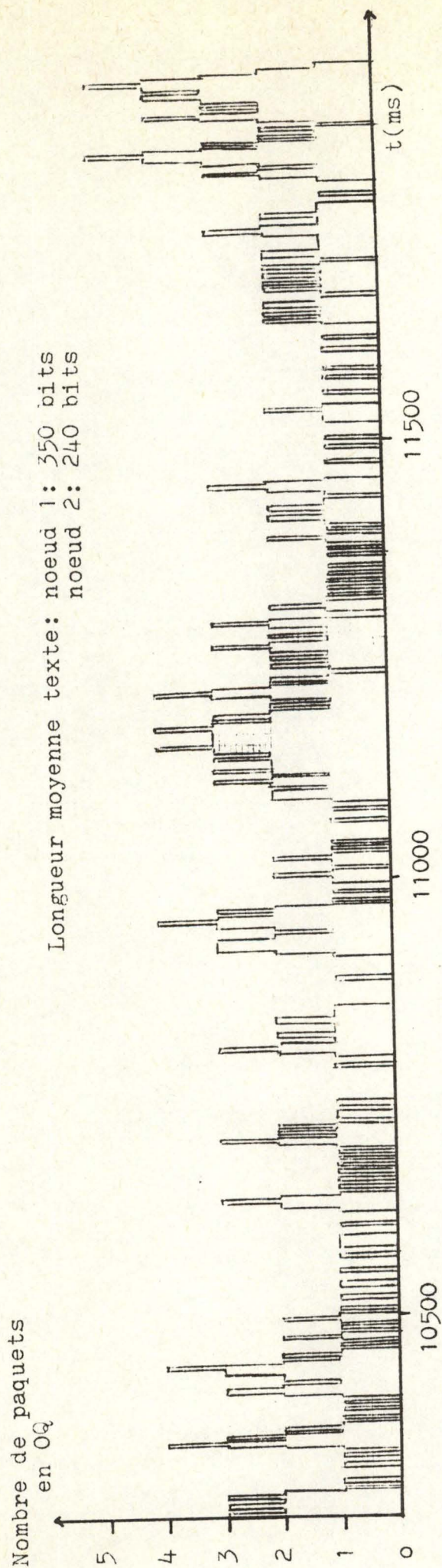


Fig 4.11. Evolution du nombre de paquets en OQ (essai 3, noeud 1)

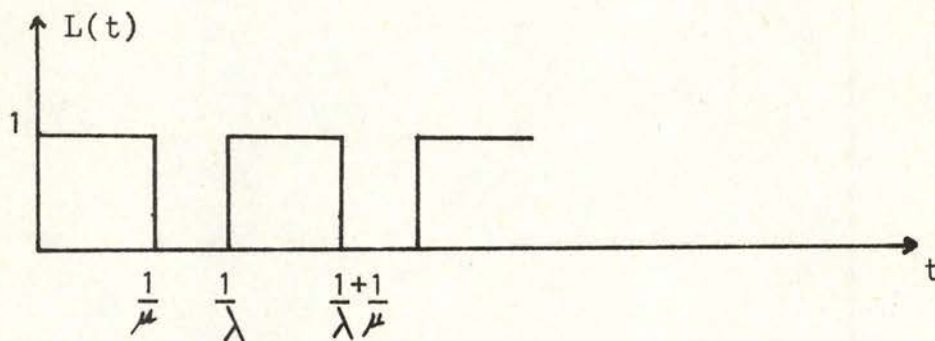


#### 4.4. Fréquence d'échantillonnage.

Nous allons parler de la fréquence d'échantillonnage des files en fonction du nombre d'unités en file, à partir de laquelle on détermine une période moyenne  $T_e$  entre deux échantillonnages, qui intervient dans la mise à jour externe de l'algorithme de routage adaptatif décrit aux chapitres précédents.

Nous examinons tout d'abord deux systèmes théoriques, le système simple D/D/1 et, le système M/M/1, le plus proche du système étudié (régime stationnaire: taux de service > taux d'arrivée  $\lambda$ ).

##### a) SYSTEME D/D/1.



$L(t)$  : longueur de la file (y compris le paquet en cours de traitement au serveur).

$L(t)$  est un signal périodique, de période  $1/\lambda$  et de valeur égale à 1 sur  $[0, 1/\mu]$

0 sur  $]1/\mu, 1/\lambda[$

$L(t)$  peut être décomposé en série de Fourier:

$$L(t) = \sum_{n=1}^{\infty} b_n \sin n\omega t + \sum_{n=0}^{\infty} a_n \cos n\omega t$$

La moyenne  $E(L(t))$  de la longueur vaut  $\frac{\lambda}{\mu} = \rho =$  taux d'occupation du serveur.

La pulsation  $\omega = \frac{2\pi}{T} = 2\pi\lambda$



Les coefficients de la série de Fourier valent respectivement

$$b_n = \frac{2}{n\pi} \sin^2 n\pi\rho$$

$$a_n = \frac{2}{n\pi} \sin n\pi\rho \cos n\pi\rho$$

$$a_0 = E(L(t)) = \frac{1}{2} \lim_{n \rightarrow 0} a_n = \rho$$

Les coefficients de la série de Fourier se comportent comme  $\frac{1}{n}$  lorsque  $n \rightarrow \infty$ , la densité spectrale de puissance décroît donc comme  $\frac{1}{n^2}$  :

$$a_n^2 + b_n^2 = \frac{4 \sin^2 n\pi\rho}{n^2 \pi^2}$$

Si l'on ne tient compte du spectre que jusqu'au premier harmonique  $n$  tel que la puissance y soit 100 fois plus faible que la puissance du 1er harmonique (ou fondamentale), on détermine  $n$  par le rapport des puissances de l'harmonique de rang  $n$  et à la fréquence nulle (régime continu):

$$\frac{4 \sin^2 n\pi\rho}{n^2 \pi^2 \rho^2} = \frac{1}{100}$$

en assimilant  $\sin^2 n\pi\rho$  à sa borne supérieure 1 et  $\pi^2$  à 10,

$$\frac{1}{n^2 \rho^2} = \frac{1}{40}$$

Si  $\rho$  augmente,  $n$  diminue et il faut échantillonner moins fréquemment, et la période  $T_e$  entre 2 échantillonnages augmente. On obtient:  $n \gg \frac{6}{\rho}$

La fréquence du signal  $L(t)$  étant  $\lambda$ , la fréquence de "coupure" vaut  $n$  fois la fréquence du signal. D'après le théorème de Shannon, il faut échantillonner au moins à une fréquence double



de la fréquence du  $n^{\text{ième}}$  harmonique, on obtient ainsi la fréquence d'échantillonnage:

$$f_e \geq 2n\lambda \qquad f_e \geq \frac{12}{\rho} \lambda = 12\mu$$

b) SYSTEME M/M/1.

Pour déterminer la période d'échantillonnage  $T_e$ , on procède comme suit.

Si  $L(t)$  est la longueur de la file, une expression de son comportement en terme de sa fonction d'auto-corrélation et de son spectre de fréquence relatif est un bon moyen de représenter son comportement dans le temps.

La fonction d'auto-corrélation  $A(t)$  de  $L(t)$  est définie par:

$$A(t) = \lim_{T \rightarrow \infty} \left( \frac{1}{T} \int_0^T n(u)n(u+t)du \right)$$

qui est une fonction du temps.

MORSE (11) montre que la transformée de Fourier de  $A(t)-L^2$ , où  $L$  est la longueur moyenne, fournit le spectre de fréquence  $W(f)$  des fluctuations de la longueur de la file autour de sa longueur moyenne:

$$W(f) \approx 4\lambda^2\mu / ((\mu-\lambda)^4 + \lambda^2(2\pi f)^2)$$

$$W(f) \approx 4\lambda\mu(\mu+\lambda)/(\mu-\lambda)^4 \qquad \text{si } f \rightarrow 0$$

$$W(f) \approx 4\mu/(2\pi f)^2 \qquad \text{si } f \rightarrow \infty$$

Si l'on veut assimiler le spectre à zéro au delà d'une fréquence  $f$  de puissance 100 fois inférieure à la puissance à la fréquence en 0, on a:

$$\frac{W(f_e)}{W(0)} \approx \frac{1}{100}$$



Ce qui donne:

$$f \approx \frac{1}{4\pi^2 \lambda^2} \left( \frac{100\lambda}{\mu + \lambda} - 1 \right) (\mu - \lambda)^4$$

D'après le théorème de Shannon, il faut échantillonner à une fréquence au moins double, d'où:

$$f_e \geq 2f$$

et on obtient une période d'échantillonnage  $T_e = \frac{1}{f_e}$

- c) Ordre de grandeur de la fréquence d'échantillonnage obtenue à partir des résultats de la simulation.

Si l'on trace des graphiques de l'évolution du nombre de paquets en file d'attente sur une période très longue, on peut observer la fréquence approximative du signal.

Les graphiques des figures 4.6, 4.6.bis, 4.10, 4.11 reproduisent cette évolution sur une certaine période. Le graphique du bas de la figure 4.6. montre l'évolution du nombre de bits sur une période.

Nous avons suggéré au chapitre 3 de baser les estimations sur une grandeur proportionnelle à la longueur des paquets: l'échantillonnage doit donc se faire en fonction de cette grandeur. Or, on constate à la figure 4.6. que l'évolution du nombre de bits en file est un signal de période approximativement égale à la période du signal correspondant de l'évolution du nombre de paquets. On peut donc approximer la fréquence d'échantillonnage du nombre de bits en file d'attente par la fréquence d'échantillonnage du nombre de paquets dans cette même file d'attente.

Les observations nous ont donnés les résultats (ordre de grandeur) fournis par le tableau 4.9. Si l'on ne tient compte du spectre (discret) que jusqu'à l'harmonique de rang  $n$  où la puissance est 100 fois plus faible que la



puissance à la fondamentale et si la puissance décroît comme  $1/n^2$ , on obtient en approximation  $n=10$ . Pour cette valeur de  $n$ , la fréquence vaut 10 fois la fréquence du signal; comme il faut échantillonner à une fréquence au moins double, on obtient:

$$f_e = 20 f$$

où  $f$  est la fréquence du signal.

Si, pour vérifier l'ordre de grandeur obtenu, on compare avec la fréquence d'échantillonnage obtenue dans le cas M/M/1, on voit au tableau 4.9. que l'ordre de grandeur est relativement bon.

taux d'arrivée	30	50	70
période observée (ms)	90	200	550
fréquence observée (hz)	11.1	5	1.8
$f_e$ (observations)	222	100	36
$T_e$	4.45	10	28
$f_e$ (M/M/1)	221	78	20
$T_e$	4.52	13	50

Tableau 4.9. (noeud 1, programme 4.2)  
(taux de service: 96.13)

Nous insistons sur le fait que ces résultats ne constituent que des ordres de grandeur. On constate que plus le taux d'occupation du serveur est élevé, plus la période d'échantillonnage est élevée.



## CHAPITRE V

### CONCLUSIONS GENERALES

Nous avons défini un type d'algorithme de routage choisi pour suivre l'évolution dynamique dans le sous-réseau de communication, au chapitre 2.

Dans le chapitre 3, nous avons discuté des modifications à apporter à un algorithme de routage du type considéré pour faire face à un certain nombre de problèmes:

- formation de boucle.
- panne de ligne.
- accessibilité d'un noeud.

Nous avons aussi proposer un mécanisme de mise à jour, pour refléter au mieux la charge moyenne sur une route, notamment par une estimation plus précise des délais en file d'attente de sortie.

- la mise à jour interne de la table des délais se fait sur une grandeur proportionnelle à la longueur des paquets: leur durée de transmission.
- la mise à jour externe de la table des délais se fait suivant 2 modes:
  - un mode immédiat si une ligne devient non opérationnelle.
  - un mode apériodique dans les autres cas.

ce mode apériodique reflète mieux les variations dues à la longueur des files de sortie que le mode périodique.

- les renseignements sur le bon fonctionnement d'une ligne et sur l'accessibilité à un noeud se trouvent dans la table des délais: les anomalies sont donc



détectées directement lors de la sélection du chemin.

Si la transmission des VDM se fait suivant le mode apériodique, ces VDM peuvent être formés à partir d'une "photographie" de la table des délais à un instant d'échantillonnage ou à partir d'une moyenne pondérée des mesures prises aux divers instants d'échantillonnage; la dépendance de cette moyenne vis à vis du passé doit être fixée. Dans les deux cas, il faut déterminer la période moyenne séparant deux instants de mesure et le seuil de variation au delà duquel la transmission des VDM sera déclenchée. On choisira la première solution si l'on veut refléter une situation la plus actuelle possible de la table des délais d'un noeud; on choisira la seconde (moyenne pondérée) si l'on veut transmettre au noeud voisin une situation sur une période plus longue, en pondérant d'une certaine façon la valeur de la mesure la plus récente intervenant dans le calcul de la moyenne.

Nous avons adopté des temps d'exécution des routines (MN, NM, "tâche principale") et des conditions de charge relativement élevés. Nous en avons déduit que la file IQ de traitement n'influence guère le comportement dynamique des autres files d'un noeud.

En utilisant la politique de transmission des ACK du protocole C, nous avons vu que la saturation de la file SQ n'est jamais atteinte, et que, lorsque le nombre maximum de paquets en  $(OQ+SQ)$  est atteint, c'est pour un nombre petit de paquets dans la SQ (1 ou 2). On peut donc, dans l'estimation des éléments de la table des délais, ne tenir compte que de la file OQ.

Nous nous sommes aussi intéressés à un ordre de grandeur de la fréquence d'échantillonnage de la file de sortie, à partir de l'analyse des résultats des simulations, spécialement du comportement, sous plusieurs conditions de charge, des files d'attente d'un noeud.

La détermination plus précise de la fréquence d'échantillonnage, à partir de l'étude de la fonction d'auto-corrélation de la longueur de la file de sortie et à partir du spectre



correspondant, sous diverses conditions de charge, pourrait constituer un prolongement de ce travail.

Un travail plus important consisterait à implémenter l'algorithme de routage, avec les modifications préconisées, sur une configuration de sous-réseau comportant un certain nombre de noeuds et de lignes.



Bibliographie

- (1) BAUWENS, Les méthodes de routage dans les réseaux à commutation de paquets, Collection des publications de la faculté des sciences appliquées de l'Université de Liège, n° 61, p. 41-69, 1976.
- (2) CEGRELL, A routing procedure for the Tidas message switching network, IEEE trans. commun., vol. COM-23, n° 6, p. 575-585, June 1975.
- (3) DANTHINE & ESCHENAUER, Influence on packet node behavior of the internode protocol, IEEE trans. commun., vol. COM-24, n° 6, p. 606-614, June 1976.
- (4) FRANK & CHOU, Routing in computer networks, Networks, vol. 1, n° 2, p. 99-112, 1972.
- (5) FULTZ & KLEINROCK, Adaptative routing techniques for store-and-forward computer-communication networks, Proc. ICC 71, p. 39.1-39.8.
- (6) HEART, KAHN, ORNSTEIN, CROWTHER & WALDEN, The interface message processor for the Arpa computer network, Afips conf. proc., vol. 36, p. 551-567, 1970.
- (7) KLEINROCK, Analytic and simulation methods in computer network design, Afips conf. proc., vol. 36, p. 569-579, 1970.
- (8) KLEINROCK, Performance models and measurement of the Arpa computer network, Proc. Online 72, p. 61-85.
- (9) MAC QUILLAN, CROWTHER, COSELL, WALDEN & HEART, Improvements in the design and performance of the Arpa network, Afips conf. proc., vol. 41, part. 2, p. 741-754, 1972.
- (10) NAYLOR, A loop-free adaptative routing algorithm for packet switched networks, Proc. Datacomm 75, p. 7.9-7.14.



- (11) MORSE, Stochastic properties of waiting lines, publication M.I.T., juin 1955.
- (12) ESCHENAUER & DANTHINE, Simulation de protocoles noeud-noeud dans une configuration de charge spécifique à Cyclades, Université de Liège, faculté des Sciences Appliquées, publication interne, 1976.
- (13) DANTHINE & MAGNEE, End-to-End protocols performance, Université de Liège, faculté des Sciences Appliquées, S.A.R.T. 76/10/10, mai 1976.



## Table des Matières.

	page
Avant-propos	1
<u>Chapitre 1: Préliminaires.</u>	
1.1. Les réseaux d'ordinateurs à commutation de paquets.	2
1.2. Le sous-réseau de communication.	9
1.2.1. Le protocole de base noeud-noeud.	10
1.2.2. La structure du noeud.	11
1.3. Introduction au comportement dynamique dans le sous-réseau.	13
<u>Chapitre 2: Le routage.</u>	
2.1. Le concept de routage.	15
2.2. Les méthodes de routage.	16
2.2.1. Classifications générale et fonctionnelle.	16
2.2.2. Le processus de décision.	17
2.2.3. Le processus de mise à jour.	18
2.2.4. Le processus d'acheminement.	18
2.2.5. Caractéristiques des principaux types de routage.	19
2.2.5.1. Les routages déterministiques.	19
2.2.5.2. Les routages adaptatifs.	19
2.3. Le routage S.Q. + Bias + P.U.	21
2.3.1. Description.	21
2.3.2. Remarque.	25



Chapitre 3: Considérations sur la dynamique des mécanismes d'échange.

3.1.	Considérations sur un routage adaptatif S.Q. + Bias + P.U.	28
3.2.	La formation de boucles - l'effet "ping-pong".	31
3.3.	La mise à jour.	34
3.4.	La mise à jour interne.	34
3.5.	La mise à jour externe.	38
3.5.1.	Calcul du VDM par une mesure instantannée.	42
3.5.1.1.	Mode de transmission immédiat	42
3.5.1.2.	Mode de transmission périodique.	43
3.5.1.3.	Mode de transmission apériodique.	44
3.5.2.	Calcul du VDM par moyenne arithmétique.	47
3.5.2.1.	Mode de transmission périodique.	47
3.5.3.	Calcul du VDM par moyenne pondérée.	49
3.5.3.1.	Mode de transmission périodique	49
3.5.3.2.	Mode de transmission apériodique	50
3.6.	L'accessibilité.	53
3.7.	Conclusions.	54



## Chapitre 4: Analyse et résultats de la simulation.

4.1.	Utilisation de la simulation	56
4.2.	Simulation entre noeuds sans trafic inverse	59
4.2.1.	Le programme de simulation.	59
	Tableaux des résultats.	64
4.2.2.	Analyse des résultats.	66
	a) le générateur de paquets.	66
	b) la file d'attente de traitement.	68
	c) la file des paquets en attente d'acquiescement.	69
	d) la file de sortie.	69
4.3.	Simulation entre 2 noeuds avec trafic inverse	74
4.3.1.	Le programme de simulation.	74
	Tableaux des résultats.	81
4.3.2.	Analyse des résultats.	83
	a) la file d'attente de traitement.	74
	b) la file des paquets en attente d'acquiescement.	86
	c) la file de sortie.	88
4.4.	Fréquence d'échantillonnage.	95

## Chapitre 5: Conclusions générales.

100



	108
<u>Bibliographie.</u>	103
<u>Table des Matières.</u>	105

## ANNEXES.

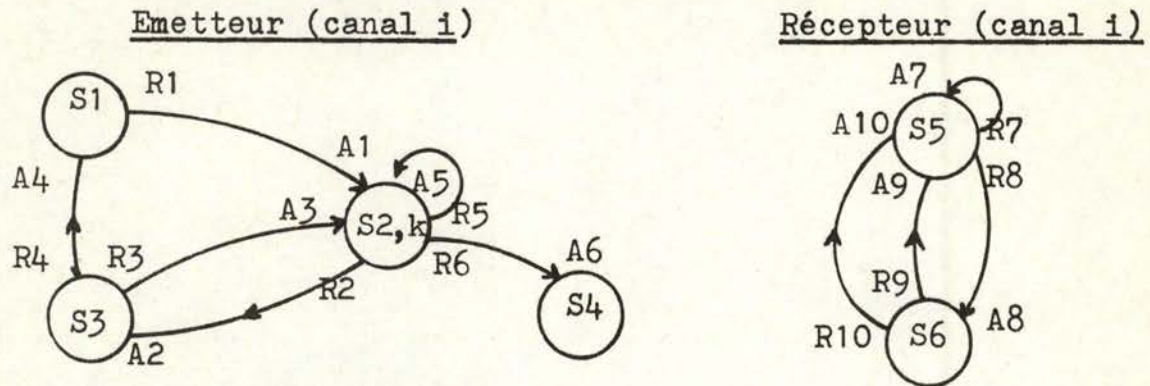
A.1. Descriptions des protocoles.	A2
A.1.1. Diagramme d'état du protocole Noeud-Noeud.	A2
A.1.2. Routine Modem-Noeud.	A4
A.1.3. Routine Noeud-Modem;	A5
A.2. Systèmes d'attente en régime stationnaire	A7
A.3. Programme de simulation du système avec trafic inverse.	A9
A.3.1. Programme (NOEUD2).	A10
A.3.2. Sous-routines de génération et de mise en file.	A21



ANNEXES

Nous décrivons dans ces annexes les protocoles utilisés dans un noeud, ainsi que le programme de simulation utilisé pour la simulation entre deux noeuds.



ANNEXE 11. Description des protocolesa) Diagramme d'état du protocole noeud-noeud

SE état de l'émetteur.

NE n° du dernier paquet émis.

NR n° du dernier paquet reçu.

Etat 1 Attente d'un paquet à envoyer sur le canal i

R1 Paquet à envoyer.

A1  $SE_i = 1$ ;  $NE_i = \overline{NE_i}$ ;  $K=0$ ; envoi du paquet avec i et  $NE_i$ ; début du time-out.

Etat 2 Attente d'un ACK

R2 Réception d'un ACK pour le canal i ( $NR_i$ )

A2  $\lambda$ .

Etat 3 Analyse de l'ACK

R3  $NE_i \neq NR_i$ .

A3  $\lambda$ .

R4  $NE_i = NR_i$ .

A4  $SE_i = 0$ ; copie du paquet supprimée.

R5 Time-out avec  $K < R$ .



- A5      $K = K+1$ ; envoyer le paquet avec  $i$  et  $NE_i$ ; début du time-out.  
R6     Time-out avec  $K = R$ .  
A6      $\lambda$ .

Etat 4    Procédure de recouvrement d'erreur (ERP)

Etat 5    Attente d'un paquet sur le canal  $i$ .

- R7     CRC faux.  
A7      $\lambda$ .  
R8     CRC correct, paquet sur canal  $i$ .  
A8     contrôle de l'initialisation.

Etat 6    Contrôle de  $NE_i$

- R9      $NE_i = NR_i$   
A9     paquet écarté (double).  
R10     $NE_i \neq NR_i$   
A10     $NR_i = \overline{NR_i}$ ; envoyer  $NR_i$  comme ACK.



b) Routine Modem-Noeud

Cette routine, qui est conduite par les interruptions générées par les modems input à la fin de transfert DMA, peut être décrite comme suit:

```

IF CRC correct
  THEN traiter le bloc ACK
    IF paquet de contrôle ou double
      THEN allocation spéciale
      ELSE IF FBL non vide
        THEN modifier le bit de parité du canal;
          positionner le flag NRR;
          BUFIN MODEM MTQ;
          allocation normale
        ELSE allocation spéciale
      ELSE allocation spéciale

```

"traiter le bloc ACK" signifie que ce bloc ACK est comparé au NER (registre contenant les numéros des derniers paquets émis sur chaque canal). Une identité bit-à-bit libère le canal correspondant ( $SE_i = P$ ) et renvoie le buffer de la  $SQ_i$  dans la FBL.

"allocation spéciale" signifie que le même buffer est réalloué au modem et une nouvelle opération DMA est initialisée.

"allocation normale" signifie que le FBL fournit le nouveau buffer alloué au modem avant une nouvelle initialisation DMA.

NRR est un registre avec les numéros des derniers paquets reçus sur chaque canal.

SER est un registre avec l'état de chaque canal.



c) Routine Noeud-Modem

Cette routine est conduite par les interruptions générées par les modems output à la fin d'un transfert DMA. Sa tâche principale est d'envoyer sur la ligne libre un nouveau paquet et de transférer le paquet qui vient d'être envoyé dans la SQ. Plusieurs protocoles peuvent être envisagés.

Protocole A.

```
IF canaux non tous occupés
  THEN IF OQ non vide
    THEN envoyer un paquet OQ
    ELSE IF SQ non vide
      THEN envoyer un paquet SQ
      ELSE envoyer un paquet de contrôle
    ELSE envoyer un paquet SQ
```

S'il n'y a pas de nouveau paquet à envoyer, un paquet déjà envoyé mais non acquitté est réémis sans attendre la condition de TIME-OUT. C'est seulement si l'OQ et la SQ sont vides qu'un paquet de contrôle est émis.

La retransmission des paquets de la SQ lorsque l'OQ est vide est critiquable: un paquet entrant dans l'OQ doit attendre la fin de la retransmission avant d'être envoyé, et ce protocole introduit un délai supplémentaire spécialement sous des conditions de charge faible ou moyenne. Ce problème est éliminé par la modification suivante.

Protocole B.

```
IF canaux non tous occupés
  THEN IF OQ non vide
    THEN envoyer un paquet OQ
    ELSE IF flog NRR positionné
      THEN envoyer un paquet de contrôle
      ELSE RETURN
    ELSE IF flag NRR positionné
      THEN envoyer un paquet de contrôle
      ELSE RETURN
```



Dans ce cas, la retransmission des paquets SQ est seulement basée sur le TIME-OUT. Le modem peut être inoccupé et dans ce cas, la routine noeud-modem doit être éveillé si le flag est positionné, ou si un nouveau paquet entre dans l'OQ ou si un ACK est reçu.

Comme les protocoles A et B placent les blocs ACK dans des paquets réguliers du trafic inverse, l'émetteur doit attendre la fin de la réception avant d'avoir accès au bloc ACK. Pour libérer les buffers de la SQ le plus tôt possible, un autre protocole peut être utilisé.

### Protocole C.

```
IF flag NRR positionné
  THEN envoyer un paquet de contrôle
  ELSE IF canaux non tous occupés
    THEN IF OQ non vide
      THEN envoyer un paquet OQ
      ELSE RETURN
    ELSE RETURN
```

Ici, la priorité est donnée à la transmission du bloc ACK par un paquet de contrôle même si des paquets attendent dans l'OQ. On veut libérer les buffers le plus tôt possible et le prix payé est une réduction de la capacité utile de la ligne, due à un overhead hardware supplémentaire des paquets de contrôle.



ANNEXE 2Systèmes d'attente en régime stationnaire.Valeurs caractéristiques.

$k$  : taux d'arrivée (processus d'arrivée de Poisson)  $k < 1$

$\bar{s}$  : temps de service moyen.

$g$  : taux d'utilisation du serveur  $g = k \bar{s}$

$\bar{n}$  : nombre moyen d'unités dans le système.

$\bar{u}$  : temps moyen de présence dans le système.

$\bar{f}$  : nombre moyen d'unités en file d'attente.

$\bar{w}$  : temps moyen d'attente.

$CV_s$  : coefficient de variation du temps de service (moyenne/écart-type).

Système M/G/1.

$$\bar{n} = k + \frac{k^2}{2(1-k)} (1 + CV_s^2)$$

$$\bar{u} = \bar{s} + \frac{\bar{s}k}{2(1-k)} (1 + CV_s^2)$$

$$\bar{f} = \frac{k^2}{2(1-k)} (1 + CV_s^2)$$

$$\bar{w} = \frac{\bar{s}k}{2(1-k)} (1 + CV_s^2)$$

Système M/M/1: temps de service de distribution exponentielle

$$\bar{n} = \frac{k}{1-k}$$

$$\bar{f} = \frac{k^2}{1-k}$$

$$\bar{u} = \frac{\bar{s}}{1-k}$$

$$\bar{w} = \frac{\bar{s}k}{1-k}$$



Système M/G/1: temps de service constant.

$$\bar{n} = \frac{k(1-k/2)}{1-k}$$

$$\bar{f} = \frac{k^2}{2(1-k)}$$

$$\bar{u} = \frac{\bar{s}(1-k/2)}{1-k}$$

$$\bar{w} = \frac{\bar{s}k}{2(1-k)}$$



ANNEXE 3.PROGRAMME DE SIMULATIONDU SYSTEME AVEC TRAFIC INVERSE

Dans cette annexe, nous reproduisons le texte du programme de simulation étudié au paragraphe 4.3, écrit en FORTRAN standard.

Ce programme s'appelle NOEUD2.

Nous y avons joint les diverses sous-routines de génération et de mise en file:

GENER (X,N,IXO)

EXPO (X,Y,N,NDEP1,EMOY)

GLONG (Z,L,N,NDEP2,LMOY,LMAX,LPV,LOVER)

FILEN (LFIL,I,NP,NUM)

FILEI (LFIL,I,IA,NP,NUM,H)

## PROGRAM NOEUD2

A10

DECLARATIONS ET ZONES COMMUNES

```

COMMON HAP(1100,2),HGA(1100,2)
DIMENSION Z(2000),COM(1100),LON(1100),AR(1100)
DIMENSION LONPA(1100,2),DIQ(1100,2),HFT(1100,2),DOQ(1100,2)
DIMENSION HAC(1100,2),DEM(1100,2),LIQ(100,2),LOQ(100,2)
DIMENSION LOV(100,2)
DIMENSION NPP(2),NIN(2),NOUT(2),NOUV(2),HFIN(2),HFEM(2)
DIMENSION NSQ(2),MAXSQ(2),IS(2),SDIQ(2)
DIMENSION SCDIQ(2),SDOV(2),SCDOV(2),SDOQ(2),SCDOQ(2),SDEM(2)
DIMENSION SCDEM(2),MDIQ(2),EDIQ(2),MDOQ(2),EDOQ(2),MDOV(2)
DIMENSION EDQV(2),MDEM(2),EDEM(2)
DIMENSION HARR(2),SDSQ(2),SCDSQ(2),MDSQ(2),EDSQ(2)
DOUBLE PRECISION COM,Z
REAL MDIQ,MDEM,MDOV,MDOQ,MDSQ

```

PREMIERE PARTIE INITIALISATIONLECTURE DES PARAMETRES

```

READ (5,999) LMOY1,LMOY2,TAUX1,TAUX2,TPN,TEI

```

DONNEES

```

VLI=48.0
LPV=80
LBMAX=1024
LOVER=152
LSI=1000
NOM=1100
NDEP1=7919
NDEP2=4111
NDEP3=5557
NDEP4=6011
AMOY1=1000.0/TAUX1
AMOY2=1000.0/TAUX2
DEMV=LPV/VLI

```

```

capacité de la ligne          (kb/s)
longueur d'un paquet vide    (bits)
longueur maximum du texte    (bits)
longueur constant de (h+s)   (bits)
portée de la simulation      (paq.)
nombre max de paquets
    arrivant à chaque noeud
nombre premier utilisé
    dans chaque générateur
temps interarrivée des
    paquets aux noeuds 1 et 2 (ms)
durée de transmission d'un
    paquet vide                (ms)

```

INITIALISATION DES VARIABLES

```

DERN=0.0
DO 100 I=1,2
IS(I)=0
NPP(I)=1
NIN(I)=0
NOUT(I)=0
NOUV(I)=0

```

```

dernier temps significatif
    calculé                    (ms)
ind. de saturation de SQ(i)
n° prochain paquet arrivant en i
pointeur sur le 1er paq. de IQ(i)
pointeur sur le 1er PN en OQ(i)
pointeur sur le 1er PV en OQ(i)

```



```

51      NSQ(I)=0      | nombre de paquets dans la file SQ(i)
52      MAXSQ(I)=0    | nombre max de paq dans SQ(i)
53      SDIQ(I)=0.0    | somme et somme des carrés des durées
54      SCDIQ(I)=0.0    | d'attente en IQ(i) (ms)
55      SDOV(I)=0.0    | somme et somme des carrés des durées
56      SCDOV(I)=0.0    | d'attente des PV en OQ(i) (ms)
57      SDOQ(I)=0.0    | somme et somme des carrés des durées
58      SCDOQ(I)=0.0    | d'attente des PN en OQ(i) (ms)
59      SDEM(I)=0.0    | somme et somme des carrés des durées
60      SCDEM(I)=0.0    | de transmission des PN (ms)
61      SDSQ(I)=0.0    | somme des carrés et somme des durées
62      SCDSQ(I)=0.0    | en SQ(i) (ms)
63      HFIN(I)=0.0    | heure de la dernière fin de trait en i
64      HFEM(I)=0.0    | heure de la dernière fin de trans en i
65      100 CONTINUE
66      DO 110 I=1,100
67      DO 110 J=1,2
68      LIQ(I,J)=0      | ième PN de la file IQ du noeud j
69      LOQ(I,J)=0      | ième PN de la file OQ du noeud j
70      LOV(I,J)=0      | ième PV de la file OQ du noeud j
71      110 CONTINUE
72      DO 120 I=1,1100
73      AR(I)=0.0        | vecteurs de travail intermédiaires
74      LON(I)=0
75      DO 120 J=1,2
76      DIQ(I,J)=0.0    | durée d'attente du ième PN en IQ(j)
77      HFT(I,J)=0.0    | heure de fin de trait du PNi en j
78      DOQ(I,J)=0.0    | durée d'attente du ième PN en OQ(j)
79      HGA(I,J)=0.0    | heure de génération de l'ACK du PNi en j
80      HAC(I,J)=0.0    | heure d'acquit. du PNi en j
81      120 CONTINUE
82 C
83 C CALCUL DES HEURES D'ARRIVEE ET DES LONGUEURS DE PAQUETS AUX 2 NOEUDS
84 C
85      CALL EXP0(COM,AR,NOM,NDEP1,AMOY1)
86      DO 130 I=1,NOM
87      HAP(I,1)=AR(I)    | heure d'arrivée du PNi au noeud j
88      130 CONTINUE      | j=1
89      CALL EXP0(COM,AR,NOM,NDEP3,AMOY2)
90      DO 140 I=1,NOM
91      HAP(I,2)=AR(I)    |
92      140 CONTINUE      | j=2
93      CALL GLONG(Z,LON,NOM,NDEP2,LMOY1,LBMAX,LPV,LOVER)
94      DO 150 I=1,NOM
95      LONPA(I,1)=LON(I) | longueur du PNi arrivant au noeud j
96      DEM(I,1)=LON(I)/VLI | durée d'émission du PNi en j j=1
97      150 CONTINUE
98      CALL GLONG(Z,LON,NOM,NDEP4,LMOY2,LBMAX,LPV,LOVER)
99      DO 160 I=1,NOM
100     LONPA(I,2)=LON(I) |

```

j=2



```
01      DEM(I,2)=LON(I)/VLI
02 160 CONTINUE
03      DO 170 I=1,LSI
04      DO 170 J=1,2
05      SDEM(J)=SDEM(J)+DEM(I,J)
06      SCDEM(J)=SCDEM(J)+DEM(I,J)**2
07 170 CONTINUE
08      DO 180 I=1,2
09      MDEM(I)=SDEM(I)/LSI
10      EDEM(I)=SQRT(SCDEM(I)/LSI-MDEM(I)**2)
11 180 CONTINUE
```

A12

moyenne et ec-type des  
durées de transmission  
des PN au noeud i

```
12 C
13 C
14 C DEUXIEME PARTIE SIMULATION
15 C -----
16 C
17 C
```

18 C 1) MISE EN FILE L/Q DE TRAITEMENT DU PAQUET SUIVANT

```
19 C
20 200 NPP1=NPP(1)
21      NPP2=NPP(2)
22      IF(NPP1.LE.NOM) GOTO 210
23      IF(NPP2.LE.NOM) GOTO 220
24      PRINT 990
25      GOTO 790
26 210 IF(NPP2.GT.NOM) GOTO 230
27      IF(HAP(NPP1,1).LT.HAP(NPP2,2)) GOTO 230
28 220 I=2
29      IA=1
30      H=HAP(NPP2,2)
31      CALL FILEI(LIQ,I,IA,NIN,NPP2,H)
32      GOTO 240
33 230 I=1
34      IA=2
35      H=HAP(NPP1,1)
36      CALL FILEI(LIQ,I,IA,NIN,NPP1,H)
37 240 NPP(I)=NPP(I)+1
38      GOTO 290
```

139 C
140 C 2) MISE EN FILE LIQ DE TRAITEMENT DES PAQUETS ARRIVES AUX 2 NOEUDS  
141 C A L'INSTANT D U DERNIER TEMPS SIGNIFICATIF CALCULE  
142 C

```
143 250 I=1
144      IA=2
145 260 MPP=NPP(I)
146      IF(MPP.GT.NOM) GOTO 270
147      IF(HAP(MPP,I).GT.DERN) GOTO 270
148      H=HAP(MPP,I)
149      CALL FILEI(LIQ,I,IA,NIN,MPP,H)
150      NPP(I)=NPP(I)+1
```



A13

```
51      GOTO 260
52 270 IF(I.EQ.2) GOTO 290
53      I=2
54      IA=1
55      GOTO 260
56 C
57 C 3) RECHERCHE DU PROCHAIN PAQUET A TRAITER OU A EMETTRE
58 C   EXAMINER POUR CELA L'HEURE MINIMUM PARMI LES HEURES D'ENTREE DU
59 C   PREMIER PAQUET DE CHAQUE FILE DES 2 NOEUDS
60 C
61 C   LE PROCHAIN TRAITEMENT CONCERNE LE PAQUET EN FILE DONT L'HEURE
62 C   SIGNIFICATIVE ACTUELLE EST LA PLUS PETITE
63 C
64 290 INDI=0
65      I=1
66      IA=2
67      TMIN=0.0
68 300 IF(NIN(I).LE.0) GOTO 350
69      NINI=NIN(I)
70      M=LIQ(NINI,I)
71      IF(M.GT.2000) GOTO 320
72      IF(TMIN.EQ.0.0) GOTO 310
73      IF(HAP(M,I).GT.TMIN) GOTO 350
74 310 TMIN=HAP(M,I)
75      IND=1
76      GOTO 340
77 320 M=M-2000
78      IF(TMIN.EQ.0.0) GOTO 330
79      IF(HGA(M,IA).GT.TMIN) GOTO 350
80 330 TMIN=HGA(M,IA)
81      IND=0
82 340 INDI=1
83      INDN=I
84      INDA=IA
85      NU=M
86 C
87 C SI LA SQ D'UN NOEUD EST SATUREE, NE PAS ENVISAGER LA FILE LOQ
88 C
89 350 IF(IS(I).EQ.1) GOTO 370
90      IF(NOUT(I).LE.0) GOTO 370
91      NOUTI=NOUT(I)
92      M=LOQ(NOUTI,1)
93      IF(TMIN.EQ.0.0) GOTO 360
94      IF(HFT(M,I).GT.TMIN) GOTO 370
95 360 TMIN=HFT(M,I)
96      IND=1
97      INDI=2
98      INDN=I
99      INDA=IA
200      NU=M
```

A14

```
201 370 IF(NOUV(I).LE.0) GOTO 390
202     NOUVI=NOUV(I)
203     M=LOV(NOUVI,I)
204     IF(TMIN.EQ.0.0) GOTO 380
205     IF(HGA(M,IA).GT.TMIN) GOTO 390
206 380 TMIN=HGA(M,IA)
207     IND=0
208     INDI=2
209     INDN=I
210     INDA=IA
211     NU=M
212 390 IF(I.EQ.2) GOTO 400
213     I=2
214     IA=1
215     GOTO 300
```

---

4) TRAITEMENT CORRESPONDANT AU PAQUET DONT L'HEURE SIGNIFICATIVE  
EST LA PLUS PETITE (TMIN)

---

LES INDICATEURS IND, INDI, INDN, INDA FOURNISSENT RESPECTIVEMENT LE TYPE DE PAQUET, LE TYPE DE TRAITEMENT, LE NUMERO DU NOEUD ET LE NUMERO DE L'AUTRE NOEUD CONCERNES PAR CE TRAITEMENT

4.1) SI TOUTES LES FILES SONT VIDES, METTRE LE PROCHAIN PAQUET EN IQ

---

```
227 400 IF(INDI.EQ.0) GOTO 200
```

---

4.2) TRAITEMENT D' UN PAQUET EN LIQ

EXAMINER SON TYPE (IND=1 GENERE, SINON AUTRE NOEUD  
CALCULER SON HEURE DE FIN DE TRAITEMENT ET SA DUREE  
EVENTUELLE EN FILE

```
234 IF(INDI.NE.1) GOTO 500
235 I=INDN
236 IA=INDA
237 IF(TMIN.GT.HFIN(I)) GOTO 450
238 IF(IND.EQ.0) GOTO 430
239 HFT(NU,I)=HFIN(1)+TPN
240 IF(NU.GT.LSI) GOTO 420
241 DIQ(NU,I)=HFIN(1)-HAP(NU,I)
242 SDIQ(I)=SDIQ(I)+DIQ(NU,I)
243 SCDIQ(I)=SCDIQ(1)+DIQ(NU,I)**2
244 420 CALL FILEN(LOQ,I,NGUT,NU)
245 430 HFIN(I)=HFIN(I)+TPN
246     DERN=HFIN(I)
247     NIN(I)=NIN(I)-1
248     GOTO 700
249 450 IF(IND.EQ.0) GOTO 470
250     HFT(NU,I)=HAP(NU,I)+TPN
```



A15

```

51 CALL FILEN(LOQ,I,NOUT,NU)
52 HFIN(I)=HFT(NU,I)
53 GOTO 480
54 470 HFIN(I)=HGA(NU,IA)+TPN
55 480 DERN=HFIN(I)
56 NIN(I)=NIN(I)-1
57 GOTO 700

```

#### 4.3) EMISSION D' UN PAQUET DE LOQ

```

58 C
59 C
60 C
61 500 I=INDN
62 IA=INDA
63 IF(IND.EQ.1) GOTO 550

```

##### 4.3.1) EMISSION D' UN PAQUET VIDE SI IND=0

CALCULER L' HEURE D' ACQUITTEMENT ET ENLEVER LE PAQUET DE LA FILE SQ

```

64 C
65 C
66 C
67 C
68 C
69 IF(TMIN.GE.HFEM(I)) GOTO 530
70 HAC(NU,IA)=HFEM(I)+DEM+TEI
71 IF(NU.GT.LSI) GOTO 520
72 DOV=HFEM(I)-HGA(NU,IA)
73 SDOV(I)=SDOV(I)+DOV
74 SCDOV(I)=SCDOV(I)+DOV*DOV
75 520 HFEM(I)=HFEM(I)+DEM
76 DERN=HFEM(I)
77 NSQ(IA)=NSQ(IA)-1
78 NOUV(I)=NOUV(I)-1
79 IS(IA)=0
80 GOTO 700
81 530 HAC(NU,IA)=HGA(NU,IA)+DEM+TEI
82 HFEM(I)=HGA(NU,IA)+DEM
83 DERN=HFEM(I)
84 NSQ(IA)=NSQ(IA)-1
85 NOUV(I)=NOUV(I)-1
86 IS(IA)=0
87 GOTO 700

```

##### 4.3.2) EMISSION PROBABLE D'UN PAQUET NORMAL

```

88 C
89 C
90 C
91 550 IF(TMIN.GE.HFEM(I)) GOTO 650

```

##### 4.3.2.1) REGARDER SI UN PV PRIORITAIRE EN FILE DOIT ETRE EMIS

```

92 C
93 C
94 C
95 INDK=0
96 IF(NOUV(I).LE.0) GOTO 625
97 NOUVI=NOUV(I)
98 K=LOV(NOUVI,I)
99 IF(HGA(K,IA).GE.HFEM(I)) GOTO 600
100 HAC(K,IA)=HFEM(I)+DEM+TEI

```



A16

```

01      IF(K.GT.LSI) GOTO 570
02      DOV=HFEM(I)-HGA(K,IA)
03      SDOV(I)=SDOV(I)+DOV
04      SCDOV(I)=SCDOV(I)+DOV*DOV
05  570  HFEM(I)=HFEM(I)+DEMV
06      DERN=HFEM(I)
07      NSQ(IA)=NSQ(IA)-1
08      NOUV(I)=NOUV(I)-1
09      IS(IA)=0
10      GOTO 700
11 C
12  600  IF(INDK.EQ.1) GOTO 655
13 C
14 C   4.3.2.2) TRAITEMENT D'UN PAQUET NORMAL EN LOQ QUI A DU ATTENDRE
15 C
16      IF(NSQ(I).NE.8) GOTO 610
17      IS(I)=1
18      GOTO 700
19  610  IS(I)=0
20      HGA(NU,I)=HFEM(I)+DEM(NU,I)+TEI
21      IF(NU.GT.LSI) GOTO 620
22      DOQ(NU,I)=HFEM(I)-HFT(NU,I)
23      SDOQ(I)=SDOQ(I)+DOQ(NU,I)
24      SCDOQ(I)=SCDOQ(I)+DOQ(NU,I)**2
25  620  HFEM(I)=HFEM(I)+DEM(NU,I)
26      DERN=HFEM(I)
27      NSQ(I)=NSQ(I)+1
28      IF(NSQ(I).GT.MAXSQ(I)) MAXSQ(I)=NSQ(I)
29      NOUT(I)=NOUT(I)-1
30      NUS=NU+2000
31      H=HGA(NU,I)
32      CALL FILEI(LIQ,IA,I,NIN,NUS,H)
33      CALL FILEN(LOV,IA,NOUV,NU)
34      GOTO 700
35 C
36 C   4.3.2.3) REGARDER SI UN PV PRIORITAIRE NON ENCORE EN FILE NE DOIT
37 C   PAS ETRE EMIS. IL SERA EMIS S' IL EST GENERE AVANT LA FIN
38 C   D'EMISSION PRECEDENTE SUR CETTE DIRECTION DE LA LIGNE OU
39 C   S' IL EST GENERE APRES LA FIN D'EMISSION MAIS AVANT
40 C   L' HEURE D' ENTREE EN LOQ DU PAQUET NORMAL SUIVANT SUR
41 C   CETTE DIRECTION. LI FAUT FORCER DANS LE PROGRAMME
42 C   L' EMISSION OU LE TRAITEMENT ET L' EMISSION SUR L' AUTRE
43 C   DIRECTION DE LA LIGNE DONNANT NAISSANCE A CE PAQUET VIDE
44 C   EN S' ASSURANT TOUTEFOIS QU' UN PAQUET VIDE N' EST PAS LUI
45 C   NON PLUS PRIORITAIRE POUR L' EMISSION SUR CETTE AUTRE
46 C   DIRECTION.
47 C
48  625  IF(NSQ(IA).EQ.8) GOTO 600
49      IF(NOUT(IA).LE.0) GOTO 640
50  626  NIA=NOUT(IA)

```



A17

```
351      NI=LOQ(NIA,IA)
352      EST=HFT(NI,IA)+DEM(NI,IA)+TEI
353      IF(EST.GT.HFEM(I)) GOTO 600
354 627 IF(NOUV(IA).LE.0) GOTO 630
355      NVIA=NOUV(IA)
356      NVI=LOV(NVIA,IA)
357      IF(HGA(NVI,I).GT.HFEM(IA)) GOTO 629
358      HAC(NVI,I)=HFEM(IA)+DEMV+TEI
359      IF(NVI.GT.LSI) GOTO 628
360      DOV=HFEM(IA)-HGA(NVI,I)
361      SDOV(IA)=SDOV(IA)+DOV
362      SCDOV(IA)=SCDOV(IA)+DOV*DOV
363 628 NSQ(I)=NSQ(I)-1
364      NOUV(IA)=NOUV(IA)-1
365      HFEM(IA)=HFEM(IA)+DEMV
366      IS(I)=0
367      GOTO 627
368 629 IF(HGA(NVI,I).GT.HFT(NI,IA)) GOTO 630
369      HAC(NVI,I)=HGA(NVI,I)+DEMV+TEI
370      NSQ(I)=NSQ(I)-1
371      NOUV(IA)=NOUV(IA)-1
372      HFEM(IA)=HAC(NVI,I)-TEI
373      IS(I)=0
374      GOTO 627
375 630 IF(HFT(NI,IA).GT.HFEM(IA)) GOTO 632
376      DOQ(NI,IA)=HFEM(IA)-HFT(NI,IA)
377      IF((EST+DOQ(NI,IA)).GT.HFEM(I)) GOTO 600
378      HGA(NI,IA)=HFEM(IA)+DEM(NI,IA)+TEI
379      IF(NI.GT.LSI) GOTO 635
380      SDOQ(IA)=SDOQ(IA)+DOQ(NI,IA)
381      SCDOQ(IA)=SCDOQ(IA)+DOQ(NI,IA)**2
382      GOTO 635
383 632 HGA(NI,IA)=EST
384 635 HFEM(IA)=HGA(NI,IA)-TEI
385      HAC(NI,IA)=HFEM(I)+DEMV+TEI
386      IF(NI.GT.LSI) GOTO 638
387      DOV=HFEM(I)-HGA(NI,IA)
388      SDOV(I)=SDOV(I)+DOV
389      SCDOV(I)=SCDOV(I)+DOV*DOV
390 638 HFEM(I)=HFEM(I)+DEMV
391      NOUT(IA)=NOUT(IA)-1
392      DERN=HFEM(I)
393      NIS=NI+2000
394      H=HGA(NI,IA)
395      CALL FILEI(LIQ,1,IA,NIN,NIS,H)
396      GOTO 700
397 640 IF(NIN(IA).LL.0) GOTO 200
398      NIA=NIN(IA)
399      NI=LIQ(NIA,IA)
400      IF(NI.LT.2000) GOTO 644
```



A18

```

01      NI=NI-2000
02      IF(HGA(NI,I).GT.HFIN(IA)) GOTO 642
03      HFIN(IA)=HFIN(IA)+TPN
04      GOTO 643
05  642 HFIN(IA)=HGA(NI,I)+TPN
06  643 NIN(IA)=NIN(IA)-1
07      GOTO 640
08  644 ESTI=HAP(NI,IA)+TPN+DEM(NI,IA)+TEI
09      IF(ESTI.GT.HFEM(I)) GOTO 600
10      IF(HAP(NI,IA).GT.HFIN(IA)) GOTO 645
11      DIQ(NI,IA)=HFIN(IA)-HAP(NI,IA)
12      IF((ESTI+DIQ(NI,IA)).GT.HFEM(I)) GOTO 600
13      HFT(NI,IA)=HFIN(IA)+TPN
14      IF(NI.GT.LSI) GOTO 648
15      SDIQ(IA)=SDIQ(IA)+DIQ(NI,IA)
16      SCDIQ(IA)=SCDIQ(IA)+DIQ(NI,IA)**2
17      GOTO 648
18  645 HFT(NI,IA)=HAP(NI,IA)+TPN
19  648 HFIN(IA)=HFT(NI,IA)
20      NIN(IA)=NIN(IA)-1
21      DERN=HFIN(IA)
22      CALL FILEN(LOQ,IA,NOUT,NI)
23      GOTO 626
    
```

```

24 C
25 650 INDK=1
26      GOTO 625
    
```

```

27 C
28 C 4.3.2.4) TRAITEMENT DIRECT D' UN PAQUET NORMAL EN LOQ, SANS ATTENTE
29 C
    
```

```

30 655 IF(NSQ(I).NE.8) GOTO 659
31      IS(I)=1
32      GOTO 700
33 659 IS(I)=0
34      HGA(NU,I)=HFT(NU,I)+DEM(NU,I)+TEI
35      HFEM(I)=HFT(NU,I)+DEM(NU,I)
36      DERN=HFEM(I)
37      NSQ(I)=NSQ(I)+1
38      IF(NSQ(I).GT.MAXSQ(I)) MAXSQ(I)=NSQ(I)
39      NOUT(I)=NOUT(I)-1
40      NUS=NU+2000
41      H=HGA(NU,I)
42      CALL FILEI(LIQ,IA,I,NIN,NUS,H)
43      CALL FILEN(LOV,IA,NOUV,NU)
    
```

```

44 C
45 C CONTROLE DE FIN DE SIMULATION
46 C
    
```

```

47 700 IF(HAC(LSI,1).EQ.0.0) GOTO 250
48 C
49 C
50 C
    
```



51 C TROISIEME PARTIE CALCUL DES RESULTATS STATISTIQUES

A19

52 C -----

53 C

54 C

55 750 DO 760 I=1,2

56 MDIQ(I)=SDIQ(I)/LSI

57 EDIQ(I)=SQRT(SCDIQ(I)/LSI-MDIQ(I)\*\*2)

58 MDOV(I)=SDOV(I)/LSI

59 EDOV(I)=SQRT(SCDOV(I)/LSI-MDOV(I)\*\*2)

60 MDOQ(I)=SDOQ(I)/LSI

61 EDOQ(I)=SQRT(SCDOQ(I)/LSI-MDOQ(I)\*\*2)

moyennes et écart-type  
des durées d'attente en  
IQ, d'attente des PV en  
OQ, d'attente des PN en  
OQ, d'attente en SQ au  
noeud i.

62 760 CONTINUE

63 DO 770 I=1,LSI

64 DO 770 J=1,2

65 RINT=HAC(I,J)-HGA(I,J)+TEI

66 SDSQ(J)=SDSQ(J)+RINT

67 SCDSQ(J)=SCDSQ(J)+RINT\*RINT

68 770 CONTINUE

69 DO 780 I=1,2

70 MDSQ(I)=SDSQ(I)/LSI

71 EDSQ(I)=SQRT(SCDSQ(I)/LSI-MDSQ(I)\*\*2)

72 780 CONTINUE

73 C

74 C

75 C IMPRESSION

76 C

77 C

78 790 PRINT 850,VLI,TAUX1,TAUX2,LMOY1,LMOY2,TPN,TEI

79 DO 800 I=1,2

80 PRINT 870,I

81 PRINT 900,MDIQ(I),EDIQ(I)

82 PRINT 910,MDOV(I),EDOV(I)

83 PRINT 920,MDOQ(I),EDOQ(I)

84 PRINT 930,MDEM(I),EDEM(I)

85 PRINT 940,MDSQ(I),EDSQ(I)

86 PRINT 950,MAXSQ(I)

87 800 CONTINUE

88 PRINT 970

89 DO 820 I=1,LSI

90 PRINT 980,I,LONPA(I,1),HAP(I,1),HFT(I,1),DOQ(I,1),HGA(I,1),

91 \*HAC(I,1),LONPA(I,2),HAP(I,2),HFT(I,2),DOQ(I,2),HGA(I,2),HAC(I,2)

92 820 CONTINUE

93 C

94 C

95 850 FORMAT(1H1,1X,'VITESSE LIGNE =',F6.1,3X,'TAUX ARRIVEE EN 1 ET 2 :'

96 \*,2F6.1,/,1X,'LONG. MOY. TEXTE EN 1 ET 2 :',2I6,/,1X,'TEMPS TRAIT

97 \*EMENT ET TEMPS N-M :',2F6.1,///)

98 870 FORMAT(1X,'NOEUD ',I2,/,1X,'-----',/)

99 900 FORMAT(1X,'MOYENNE ET EC-TYPE DUREE PN GENERES:',7X,2F11.3,/)

100 910 FORMAT(1X,'MOYENNE ET EC-TYPE DUREE PV EN OQ :',7X,2F11.3,/)

```

01 920 FORMAT(1X,'MOYENNE ET EC-TYPE DUREE PN EN OQ :',7X,2F11.3,/)
02 930 FORMAT(1X,'MOYENNE ET EC-TYPE DUREE EMISSION :',7X,2F11.3,/)
03 940 FORMAT(1X,'MOYENNE ET EC-TYPE DUREE EN SQ :',7X,2F11.3,/)
04 950 FORMAT(1X,'MAXSQ =',7X,I4,///)
05 970 FORMAT(1H1,5X,'LONG',3X,'ARRIVEE',2X,'FIN TRAIT',1X,'DUREE OQ',
06 *4X,'GEN ACK',5X,'ACQUIT',1X,'*',1X,'LONG',3X,'ARRIVEE',2X,'FIN TRA
07 *IT',1X,'DUREE OQ',4X,'GEN ACK',5X,'ACQUIT',///)
08 980 FORMAT(1X,I4,1X,I4,1X,F10.1,1X,F10.1,1X,F8.1,1X,F10.1,1X,F10.1,1X,
09 *'',1X,I4,1X,F10.1,1X,F10.1,1X,F8.1,1X,F10.1,1X,F10.1,/)
10 990 FORMAT(1X,'NPP(I)>NOM',/)
11 999 FORMAT(I4,I4,F5.1,F5.1,F4.1,F5.1)
12 C
13 C
14 STOP
15 END

```



```

1      SUBROUTINE GENER(X,N,IXO)
2 C
3      DIMENSION X(N)
4      DOUBLE PRECISION X,P,PM1,ADP,B,C
5      A=16807.0
6      P=2.147483648D9
7      PM1=P-1.0
8      ADP=A/P
9      K=IDINT(ADP*IXO)
10     B=A*IXO - K*PM1
11     IF(B.GE.PM1) B=B - PM1
12     DO 10 I=1,10
13     K = IDINT(ADP*B)
14     C = A*B - K*PM1
15     IF(C.GE.PM1) C = C-PM1
16     B=C
17 10 CONTINUE
18     X(1) =B
19     DO 20 I=2,N
20     K=IDINT(ADP*X(I-1))
21     X(I) = A*X(I-1) - K*PM1
22     IF(X(I).GE.PM1) X(I)=X(I)-PM1
23 20 CONTINUE
24     DO 30 I=1,N
25     X(I) = X(I)/PM1
26 30 CONTINUE
27     RETURN
28     END

```

A21

GENER (X,N,IXO)

Cette routine produit N valeurs  $X(i), i=1, \dots, N$ , d'une variable aléatoire de distribution uniforme sur  $(0,1)$ . IXO est un nombre premier compris entre 2 et  $2^{31}-1$ . Le générateur utilisé est le générateur congruentiel multiplicatif de Lewis:

$$X_{i+1} = A X_i \text{ mod } m$$

$$X_0 = \text{IXO}$$

où A vaut  $7^5 = 16807$  et la période  $m-1$  vaut  $2^{31}-2$ . Pour chaque valeur de IXO, on obtient une génération différente de nombres aléatoires.

```

1      SUBROUTINE EXPO(X,Y,N,NDEP1,EMOY)
2 C
3      DIMENSION X(N),Y(N)
4      DOUBLE PRECISION X
5      EMOY=-EMOY
6      CALL GENER(X,N,NDEP1)
7      DO 10 I=1,N
8      X(I)=EMOY*ADLOG(X(I))
9      10 CONTINUE
10     Y(1) = SNGL(X(1))
11     DO 20 I=2,N
12     Y(I) = Y(I-1) + SNGL(X(I))
13     20 CONTINUE
14     RETURN
15     END

```

A22

EXPO (X,Y,N,NDEP1,EMOY)

Cette routine produit les N premières valeurs  $Y(i), i=1, \dots, N$ ,  
 D'un processus de Poisson de taux  $1/EMOY$ . Ces valeurs sont  
 obtenues à partir de N valeurs  $X(i), i=1, \dots, N$ , d'une variable  
 aléatoire de distribution exponentielle de moyenne EMOY qui  
 représente le temps interarrivée dans un processus d'arrivée  
 de Poisson. Les valeurs de cette variable exponentielle  
 sont obtenues à partir des valeurs d'une variable de distri-  
 bution uniforme sur (0,1) fournies par la routine GENER.



```

1      SUBROUTINE GLONG(Z,L,N,NDEP2,LMOY,LMAX,LPV,LOVER)
2 C
3      DIMENSION Z(2000),L(N)
4      DOUBLE PRECISION Z
5      DOUBLE PRECISION P1
6      M=2000
7      P1 = LMAX + 1
8      MLMOY=-LMOY
9      K=0
10     CALL GENER(Z,M,NDEP2)
11     DO 3J J=1,N
12 50    K=K+1
13     Z(K)=MLMOY*DLOG(Z(K))
14     IF(Z(K).GE.P1) GOTO 50
15     L(J)=10*INT(Z(K))+LOVER
16 80    CONTINUE
17     RETURN
18     END

```

A23

GLONG (Z,L,N,NDEP2,LMOY,LMAX,LPV,LOVER)

Cette routine produit N valeurs  $L(i), i=1, \dots, N$ , qui sont les parties entières d'une variable aléatoire  $L=LOVER + B$ , où LOVER est une constante et où B est une variable aléatoire de distribution exponentielle de moyenne LMOY tronquée en LMAX. Ces valeurs sont les longueurs LP des paquets. Les valeurs de B sont obtenues à partir des valeurs d'une variable aléatoire uniforme dans (0,1) fournies par GENER.

```

1      SUBROUTINE FILEN(LFIL,I,NP,NUM)
2 C
3      DIMENSION LFIL(100,2),NP(2)
4      NP(I)=NP(I)+1
5      MP=NP(I)
6      IF(MP.EQ.1) GOTO 50
7      LL=MP-1
8      KK=MP
9      J=1
10     20 LFIL(KK,J)=LFIL(KK-1,I)
11         KK=KK-1
12         IF(J.EQ.LL) GOTO 50
13         J=J+1
14         GOTO 20
15     50 LFIL(1,I)=NUM
16         RETURN
17         END

```

A24

FILEN (LFIL,I,NP,NUM)

Cette routine place le paquet n° NUM dans la file d'attente LFIL (.,I) du noeud I.

NP (I) est un pointeur sur le premier paquet à extraire de la file (stratégie FIFO). Le dernier paquet mis en file se trouve en LFIL (1,I), les autres sont décalés d'un rang dans la file.



```

1      SUBROUTINE FILEI(LFIL,1,IA,NP,NUM,H)
2 C
3      COMMON HAP(1100,2),HGA(1100,2)
4      DIMENSION LFIL(100,2),NP(2)
5      NP(I)=NP(I)+1
6      MP=NP(I)
7      IF(MP.EQ.1) GOTO 70
8      LL=MP-1
9      DO 30 JJ=1,LL
10     LF=LFIL(JJ,I)
11     IF(LF.GT.2000) GOTO 20
12     IF(HAP(LF,I).LT.H) GOTO 50
13     GOTO 30
14 20  LF=LF-2000
15     IF(HGA(LF,IA).LT.H) GOTO 50
16 30  CONTINUE
17     LFIL(MP,I)=NUM
18     RETURN
19 50  KK=MP
20 60  LFIL(KK,I)=LFIL(KK-1,I)
21     KK=KK-1
22     IF(KK.NE.JJ) GOTO 60
23     LFIL(KK,I)=NUM
24     RETURN
25 70  LFIL(1,I)=NUM
26     RETURN
27     END

```

A25

# FILEI (LFIL,I,IA,NP,NUM,H)

Cette routine place le paquet n° NUM et d'heure d'entrée en file H, dans la file LFIL (.,I) du noeud I.

NP(I) est le pointeur sur le premier paquet à extraire de la file. IA est le n° de l'autre noeud.

Les paquets dont les numéros sont supérieurs à 2000 viennent du noeud IA, les autres arrivent au noeud I en provenance du générateur de paquets. Les paquets n'arrivent pas nécessairement dans la file chronologiquement. Ils y sont rangés suivant un ordre décroissant de leur heure d'arrivée.

HAP (LF,I) est l'heure d'arrivée du paquet n° LF du noeud I en file d'attente de traitement (LIQ (.,I)) du noeud I.

HGA (LF,IA) est l'heure d'arrivée du paquet n° LF venant du noeud IA et arrivant en file d'attente de traitement (LIQ(.,I)) du noeud I.



